Circular no.: MCX/CTCL/502/2023 Aug 02, 2023

**New Tick By Tick Trading Market Data Interface (API) –**
**MCX Enhanced Order Book Interface (EOBI) - Version 1.2**

In terms of provisions of the Rules, Bye-Laws and Business Rules of the Exchange and in continuation to Exchange circular no. MCX/CTCL/443/2023 dated June 30, 2023, MCX/CTCL/493/2021 dated August 5, 2021 and MCX/CTCL/489/2022 dated August 22, 2022 Members of the Exchange are notified as under:

The Exchange is in the process of implementing a new commodity derivative platform. Pursuant to the same, there would be changes in the interfaces. Trading Members and Empanelled vendors are requested to note that, the Exchange has released new Tick By Tick Trading Market Data Interface API, which will be in effect once the new trading platform is live. Below are the interfaces details:

| Sr. No. | Interface Name | Description |
|---|---|---|
| 1. | Tick By Tick Trading Market Data Interface - MCX Enhanced Order Book Interface (EOBI) - Version 1.2 | This interface provides the entire visible order book, by publishing information on each individual order along with executions and state information in real-time and in an un-netted manner. The interface is available for a selected group of derivatives market benchmark products.<br>This interface provides greater transparency and efficiency, together with a high throughput at minimal latency. |

In case of any queries or clarification on new interfaces document, trading members/vendors are requested to get in touch on following contact details:

- Email - apisupport@mcxindia.com
- Phone: +91 22 – 6649 4040 / 6731 8888

Trading Members and Empanelled vendors are requested to take note of the same.

Abhay Angarkar
AVP - Technology
Encl.: As above

Kindly contact Customer Service Team on 022 – 6649 4040 or send an email at customersupport@mcxindia.com for any clarification.

---------- Corporate office ----------
Multi Commodity Exchange of India Limited
Exchange Square, CTS No. 255, Suren Road, Chakala, Andheri (East), Mumbai – 400 093
Tel.: 022 – 6649 4000 Fax: 022 – 6649 4151
www.mcxindia.com email: customersupport@mcxindia.com

**Multi Commodity Exchange of India Limited**

# Tick By Tick Trading Market Data Interface - MCX Enhanced Order Book Interface (EOBI)

**Version 1.2**
**July 05, 2023**

## Copyright



## Document details

| Name | Version no. | Description |
|---|---|---|
| MCX_EOBI_API | V 1.2 | API documentation for Tick By Tick Interface |

## Document Revision List

| Revision No. | Revision Date | Revision Description |
|---|---|---|
| 1 | 22-Jun-2021 | Created of base document |
| 2 | 05-Jul-2023 | Addition<br>Added Instrument Info Message in Snapshot block diagram<br>Instrument Summary (13601) :<br>- TradeCondition field is added<br>- Life time low and Life time high enums are added in >MDEntryType field<br><br>Modification<br>- Section 3.4.3 Snapshot messages; Picture 8: A snapshot cycle of a product. Added<br>- Section 4.20 Instrument Info 13603 Message description modified.<br>Deletion<br>- Instrument Summary-13601 Removed DPL enum from >MDEntryType field |

## MCX Contact Details

| The Exchange's Member & Vendor may contact Technology Division to seek clarification at: | |
|---|---|
| Multi Commodity Exchange of India Limited<br>Exchange Square, Suren Road, Chakala, Andheri (East), Mumbai 400 093.<br>www.mcxindia.com | Tel: +91 – 22 – 66494000 / 67318888<br>Fax: +91 – 22 – 66494151<br>Email – apisupport@mcxindia.com |

## Restriction on Use and Disclaimer of Information and Data

## Contents

# 1 List of Abbreviations

| Abbreviation | Description |
|---|---|
| EOBI | Enhanced Order Book Market Data Interface |
| EMDI | Enhanced Price Level aggregated Market Data Interface |
| ETI | Enhanced Trading Interface |
| FAST | FIX Adapted for STreaming (FAST Protocol) (FAST ProtocolSM ). FIX Adapted for STreaming is a standard which has been developed by the Data Representation and Transport Subgroup of FPLs Market Data Optimization Working Group. FAST uses proven data redundancy reductions that leverage knowledge about data content and data formats |
| FIX | Financial Information eXchange. The Financial Information eXchange ("FIX") Protocol is a series of messaging specifications for the electronic communication of trade-related messages. |
| In-band | Incremental and snapshots are delivered in the same channel |
| Out-of-band | Incremental and snapshots are delivered on different channels. |
| PMAP | Presence Map |
| ToB | Top of Book |
| Simple instruments | Single leg outright contracts |
| Complex instruments | Any combination of single leg outright contracts, e.g. LTP Based Spread Contracts |
| T7 | T7 trading system developed by Deutsche Börse Group |

# 2 Introduction

The **MCX T7 E**nhanced **O**rder **B**ook **I**nterface (MCX T7 EOBI) provides the entire visible order book, by publishing information on each individual order along with executions and state information in real-time and in an un-netted manner. The interface is available for a selected group of derivatives market benchmark products.

This interface provides greater transparency and efficiency, together with a high throughput at minimal latency. The MCX T7 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.
- Dissemination of incremental messages (following state changes) and all Snap-shot messages follow a publishing in sequence based on:
    - Side (bid first, offer second),
    - Price (best price first),
    - Time (highest time-priority first).

The MCX T7 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 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 without any depth restriction at the time of sending.

Multicast address and port combinations of MCX T7 EOBI are different from netted market data broadcast channel.

# 3 Service Description

## 3.1 Functional Characteristics

The MCX T7 EOBI disseminates:

- The instrument identifier, side, price, and quantity of each visible order .
- 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.
- Recovery via MCX T7 EOBI snapshots.

In order to send public market data as fast as possible, the MCX T7 EOBI publishes only very specific market information. However, participants can derive certain information themselves based on the messages sent out by the MCX T7 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 MCX T7 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 MCX T7 EOBI incremental channel. But, on the other hand, trade statistics are sent out on the MCX T7 EOBI snapshot channel for recovery purposes.

### 3.1.1 Technical Characteristics

The MCX T7 EOBI contains similar technical characteristics as the MCX T7 EMDI, such as "Live - Live" multicast, distribution mode and sequence numbering schemes. Anticipating a high load, thesize of messages is kept as small as possible.

The following are highlights of the technical characteristics of the MCX T7 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 sequence numbers are incremented per channel only. Additionally, the MarketSegmentID will be provided in the Packet Header only.
- Little Endian and basic data types are used.
- Message padding for better byte alignment.
- Recovery via MCX T7 EOBI snapshot channel as similar to MCX T7 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.

All functional and technical reference data information needed for the MCX T7 EOBI is provided in contract master.

## 3.2 Order Book Management

The MCX T7 EOBI provides an **explicit** message for each order book update by publishing the instrument identifier, side, price, displayed quantity 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 products which are enabled on MCX T7 EOBI..

An outline of the **visibility** of orders on the MCX T7 EOBI is shown below:

| Order Type | Visible in Order book |
|---|---|
| Regular Limit Order | yes |
| Triggered Order – Stop Limit Order | yes |
| Regular Order – GFD / GTC / GTD | yes |
| Market Orders | yes |
| 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 MCX T7 EOBI**

For each instrument within a product, snapshot messages can be received via the MCX T7 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 MCX T7 EOBI incremental channel. On the MCX T7 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 MCX T7 EOBI incremental channel.

To assist fine filtering and error discovery on the participant side, the MCX T7 EOBI keeps messagesin line using a multi-sequencing paradigm. It uses the following two sequencing methods: **packetsequence number** and **message sequence number**.

### Packet Sequencing

Each packet on the MCX T7 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 MCX T7 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 MCX T7 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 |
| Top Of Book | 13504 |
| Product State Change | 13300 |
| Instrument State Change | 13301 |
| Product Summary | 13600 |
| Instrument Summary | 13601 |
| Snapshot Order | 13602 |
| Heartbeat | 13001 |
| Instrument Info | 13603 |
| Index Info | 13604 |

**Table 2 - MCX T7 EOBI messages with assigned template IDs**

### 3.2.1 Building the Order Book

Messages in the MCX T7 EOBI snapshot channels are grouped by product. In order to build an initial order book, participants subscribe to the MCX T7 EOBI snapshot channel. The content of one **snapshot cycle** for one product is described in subsequent sections. 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 enableparticipant synchronization between the MCX T7 EOBI snapshot channel and the MCX T7 EOBI incrementalchannel.

While subscribed to the MCX T7 EOBI snapshot channel, participants should keep processing incoming data from the MCX T7 EOBI incremental channel. Any incoming incremental messages with a sequence number higher than the *LastMsgSeqNumProcessed* received in the snapshot messageshould be applied to the order book after the full snapshot message is processed.

The following data is provided via the MCX T7 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 MCX T7 EOBI incremental channel.

As soon as trading is in the state Continuous, all visible orders in the order book will be immediately published on the MCX T7 EOBI incremental channel.

The sequencing of the data in a snapshot cycle is based on the product identifier, the instrumentidentifier 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 sortedfrom 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 pricelevel are sorted by order time priority from the oldest to the newest order. The visible order bookis disseminated per price level in a zig-zag manner, meaning both the sides (Bid and Offer) ateach 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 areprocessed 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 accordingto 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 pricelevel. The orders denoted by B3, S2, S3 and S4 are on the second price level. The orders B4and 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.

### 3.2.2 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, side, price and displayed quantity of the order.

Information about an incoming order, that matched fully against to one or more orders in theorder 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 OrderAdd message after all associated executions.

### 3.2.3 Modifying an Order

If the time-priority, price and/or displayed quantity of an existing order changes, then an OrderModify 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.

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.

### 3.2.4 Deleting an Order

When an order is deleted, the MCX T7 EOBI will publish the instrument identifier, side, price and transaction time , i.e., the fields *SecurityID, Side, Price* and *TransactTime,*

### 3.2.5 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.

The Execution Summary message contains the instrument identifier, side, , worst price, total executed quantity, resting hidden quantity (if any) and match-time information of the incoming order.

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

The individual order execution messages should be used by participants for order book maintenance to ensure the correctness of the order book. The Execution Summary messages canbe used by participants for fast trading decisions.

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 usecase.

The order execution messages will be sent whenever a visible order is **fully** or **partially** executed at its displayed price. Each **matchstep** 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 partialexecution. The same unique identifier of the trade is made available to participants by the MCX T7 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 Limitorders, respectively.

### 3.2.6 Trade Statistics

Instrument trade statistics such as opening, closing, daily low and high prices are available via the MCX T7 EOBI snapshot messages only. They are provided to participants for recovery purposes and are published included in the Instrument Summary message on the MCX T7 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 MCX T7 EOBI incremental channel, participants can derive order book and trade statistics by combining the information received via the order and execution messages.

### 3.2.7 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*.

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

### 3.2.8 Heartbeats

Functional heartbeat messages, Heartbeat, are sent at a regular interval for less active products on the MCX T7 EOBI incremental channels. A functional heartbeat message provides the message sequence number last sent in the field *LastMsgSeqNumProcessed* to allow participants to identifypotential 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.

### 3.2.9 Recovery

Due to the unreliable nature of UDP multicast, UDP packets may be duplicated, delayed, missing, or arrive in an incorrect sequence. Participants can utilize the MCX T7 EOBI snapshot channel to obtain the corresponding lost information, i.e., rebuildthe 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 following reason:

- The field *LastMsgSeqNumProcessed* in the snapshot cycle is given on product level, so in order to synchronize the MCX T7 EOBI snapshot channel and the MCX T7 EOBI incremental channel, participants should recover for all instruments in the product.

## Participant Fail-Over

In the event of a packet loss on both (Live - Live) services of an MCX T7 EOBI channel, recovery on the participant side can be achieved by recovering the order book information via the MCX T7 EOBI snapshot channel.

The MCX T7 EOBI snapshot channel is synchronized with the MCX T7 EOBI incremental channel through the use of message sequence numbering. Participants should subscribe to the MCX T7 Order Book Snapshot channel while buffering incoming messages from the MCX T7 EOBI incremental channel. Any incoming message from the MCX T7 EOBI incremental channel with a *MsgSeqNum* higher thanthe 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 failure

A failure of a MCX T7 EOBI service for a certain *PartitionID (5948)* always leads to a full restart ofthe respective service and can be detected on an EOBI channel by following characteristics:

- The *ApplSeqNum* in the Packet Header is reset to 1.
- The *MsgSeqNum* for each product or *MarketSegment* in the Message Header is reset to 1.

When a participant receives packets on a specific multicast address (either on service A or service B) with an unexpected (lesser or equal) packet header *ApplSeqNum* (usually 1), it is advised, that the participant rebuilds his order books from the new incremental message sequence or subscribes to the MCX T7 EOBI snapshot channel again.

**Note that**, because of the unreliable nature of the UDP protocol, packets may arrive out of sequence. An application might also see packets with an *ApplSeqNum* greater or equal to theprevious *ApplSeqNum* for a specific fail-over period. Whenever an application detects an unexpected new (lesser or equal) *ApplSeqNum* on a specific multicast address with a packet header *TransactTime* $t_0$ from a *new* sender, all packets from the old sender are expected to have a packet header *TransactTime* $t < t_0$.

In certain cases of a full restart of a MCX T7 EOBI service, participants must also wait for the firstmessage after the restart to be certain that a restart was executed.

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

### 3.3 Availability of Enhanced Order Book Service

The MCX T7 EOBI is available during the entire business day between product states"Start-Of-Day" and "Post-End-Of-Day".

Table 4 below shows the information typically sent on the MCX T7 EOBI during each product state.The messages listed in the table should serve as a super-set of messages and inform participantson "what-to expect" during each product state. However, it does not state any deterministicbehaviour 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,Product Summary, Instrument Summary (incl. Trade Statistics),Heartbeat |
| Pre-Trading | Product State Change, Instrument State Change,Order Mass Delete, Product Summary, Instrument Summary (incl. Trade Statistics), Heartbeat |
| Trading | Product State Change, Instrument State Change, Add Order, Modify Order, Modify Order Same Priority, Delete Order, Partial Order Execution, Full Order Execution, Execution Summary, Heartbeat, Product Summary, Instrument Summary (incl. Trade Statistics),Snapshot Order, |
| Post-Trading | Product State Change, Instrument State Change,Order Mass Delete, Product Summary, Instrument Summary (incl. Trade Statistics), Top Of Book, 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 | - |

| Product State | Messages |
|---|---|
| Halt | Product State Change, Instrument State Change,<br>Order Mass Delete,<br>Product Summary,<br>Instrument Summary (incl. Trade Statistics) |
| Holiday | Product State Change, Instrument State Change,Product Summary,<br>Instrument Summary (incl. Trade Statistics),Heartbeat |

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

**Please note that the MCX T7 EOBI snapshot channels stop after migration of all products to "Post- End-Of-Day".**

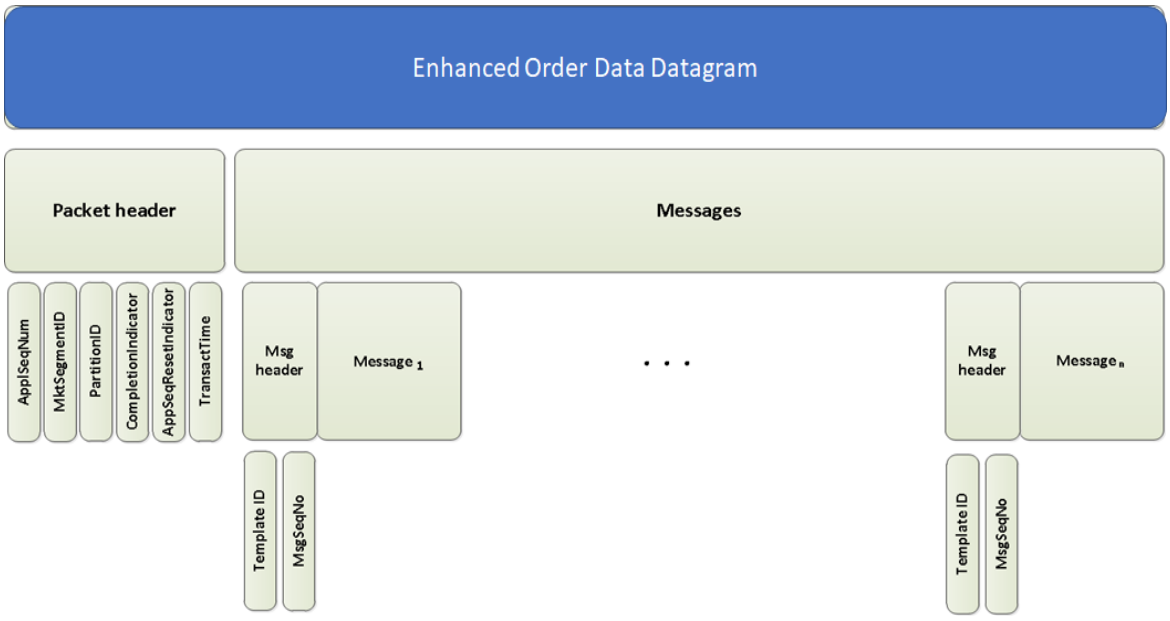
## 3.4 Message Formats

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

### 3.4.1 Datagram Structure

Each UDP datagram 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 MCX T7 EOBI follows the following structure for the datagrams sent on the network:




Picture 4: Generic Datagram structure of MCX T7 EOBI

The Packet Header in each datagram contains information about

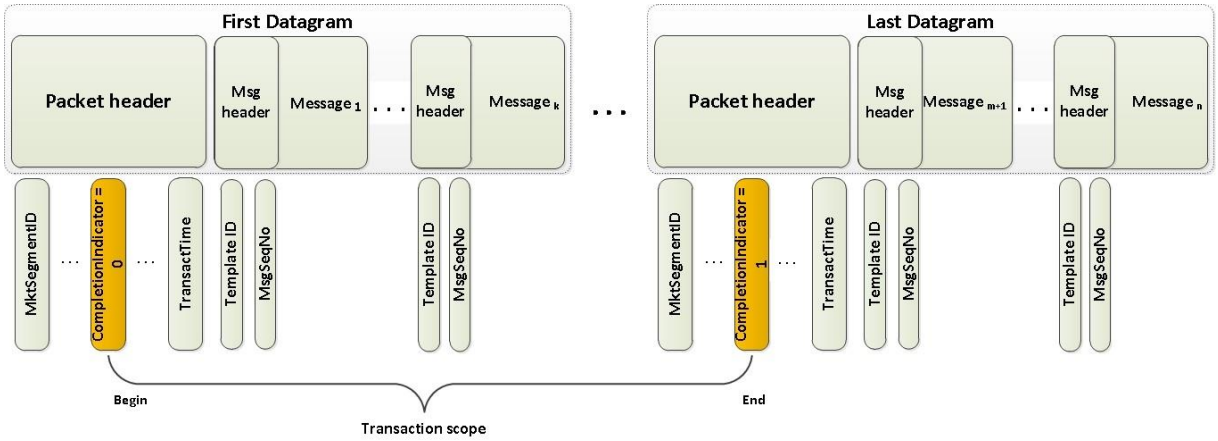
- The product and the partition ID of corresponding product,
- A contiguous packet sequence number,
- An indicator whether the **atomic unit of work** fits into one datagram,
- An indicator whether a fail-over has occurred, and
- When the packet has been sent out.

The product, *MarketSegmentID*, information can be used by participants for product filteringpurposes.

The packet sequence numbers, *ApplSeqNum*, are contiguous and are incremented per MCX T7 EOBI channel (service A and service B). They can be used by participants to detect gaps, duplicate and missing packets.

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 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 packetheader will be set to *Complete (=1)*.

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

The functional structure of each MCX T7 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 MCX T7 EOBI incremental messages are incremented per product. Message sequence numbers for the MCX T7 EOBI snapshot messages are incremented per snapshot cycle.

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

### 3.4.2 Incremental Messages

Incremental messages are sent according to the MCX T7 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 *templateID* in the message header identifies each incremental message uniquely.

MCX T7 EOBI incremental messages will be sent as long as the MCX T7 EOBI service is available. The Heartbeat messages are repeated in the configured heartbeat interval in a single datagram bysetting the message sequence number last sent to the *LastMsgSeqNumProcessed* field of the corresponding product. If the *LastMsgSeqNumProcessed* is not available, i.e., until the productstate "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 |
| Top Of Book | 13504 |
| Product State Change | 13300 |
| Instrument State Change | 13301 |
| Heartbeat | 13001 |
| Index Info | 13604 |

**Table 5 - MCX T7 Enhanced Order Book incremental messages**

For order book maintenance, the order messages Order Add, Order Modify, Order Deleteand 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 MCX 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 CrossRequest 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.

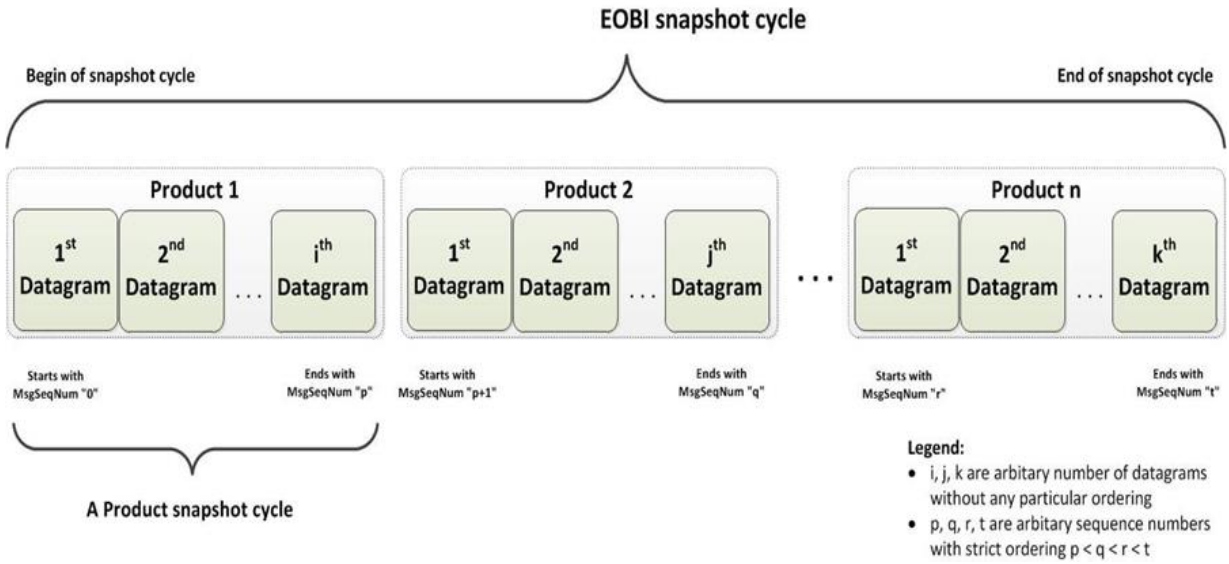
### 3.4.3 Snapshot Messages

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 contrastto MCX T7 EOBI incremental messages, MCX T7 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 MCX T7 EOBI - “Datagram Structure“.

MCX T7 EOBI snapshot messages will be sent in product states between “Start-Of-Day” and“Post-End-Of-Day”.

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



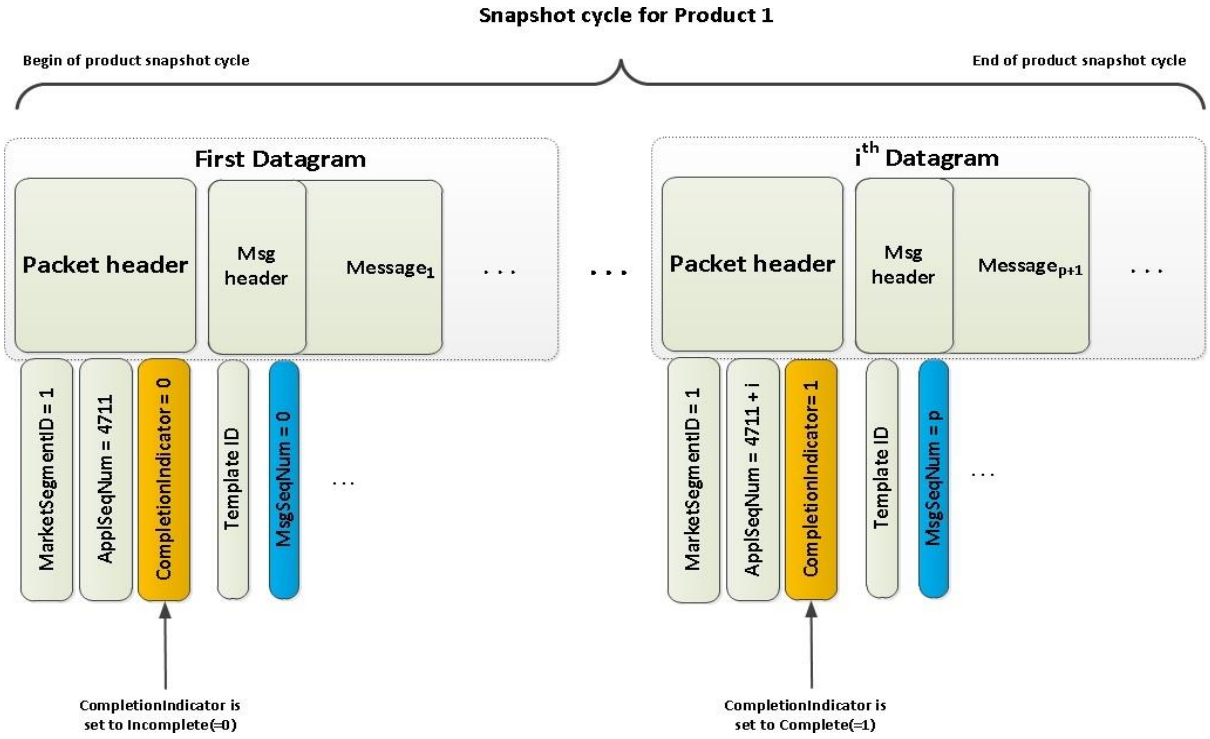

Picture 6: An overview of a snapshot cycle

It is characterized by,

- The packet sequence numbers, *ApplSeqNum,* are contiguous and areincremented across products,
- The message sequence number, *MsgSeqNum*, of the first message in the firstdatagram of a new snapshot cycle is set to zero(=0),
- The message sequence number, *MsgSeqNum,* within the same snapshot cycle isincremented 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 MCX T7 EOBI snapshot feed comprises of multiple productsnapshot 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 thepicture above.




Picture 7: The Snapshot cycle for Product1

Each message header containing the *templateID* of a message within a snapshot cycle willspecify 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 lastmessage 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[1]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 betweenincremental and snapshot feed.

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

A product has multiple instruments. The Product Summary will be given once, as it includesattributes that are identical for all instruments. However, it can include multiple InstrumentSummary 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 implicitlydefines the start of a snapshot cycle for the next product, inherently 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.

MCX T7 EOBI snapshot messages will contain order book information about the intra-day createdcomplex 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 stateuntil end of day trading state to provide participants with last available instrument[1]s BBO information.

### 3.4.4 Data Types

The following table provides an overview of the data types used in the fixed-length binary encoded messages sent out by the MCX T7 EOBI. These data types will be indicated for each field in the Chapter 8 - "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<br>2 byte signed int: 0x8000<br>4 byte signed int: 0x80000000<br>8 byte signed int: 0x8000000000000000 |
| unsigned int. | little endian byte order supported are 1, 2, 4 and 8-byteunsigned integer. | 1 byte unsigned int: 0xFF<br>2 byte unsigned int: 0xFFFF<br>4 byte unsigned int: 0xFFFFFFFF<br>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. |
| QuantityType | Quantity in integer format including 4 decimals. | 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 (00:00:00UTC on January 1st, 1970). | see 8 byte unsigned int. |

Table 6 - Data types on the MCX T7 EOBI

# 4 Functional Specification

## 4.1 Overview of Supported Message Types

The following message formats are based on:

### General

| EOBI Message | TemplateID (28500) | FIX Message | MsgType (35) |
|---|---|---|---|
| Packet Header | 13003 | MarketDataReport | U20 |
| Heartbeat | 13001 | Heartbeat | 0 |

### Trade Data

| EOBI Message | TemplateID (28500) | FIX Message | MsgType (35) |
|---|---|---|---|
| Execution Summary | 13202 | MarketDataTrade | U22 |
| Index Information | 13604 | MarketDataTrade | U22 |

### Order Data

| EOBI Message | TemplateID (28500) | FIX Message | MsgType (35) |
|---|---|---|---|
| Order Add | 13100 | MarketDataOrder | U21 |
| Top of Book | 13504 | MarketDataInstrument | U23 |

| EOBI Message | TemplateID (28500) | FIX Message | MsgType (35) |
|---|---|---|---|
| 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 |

State Change

| EOBI Message | TemplateID (28500) | FIX Message | MsgType (35) |
|---|---|---|---|
| Product State Change | 13300 | TradingSessionStatus | h |
| Mass Instrument State Change | 13302 | SecurityMassStatus | CO |
| Instrument State Change | 13301 | SecurityStatus | f |

Snapshot

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

## 4.2 Packet Header = 13003

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 incase 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. |

| Tag | Field Name | Req'd | Len | Ofs | Data Type | Description |
|---|---|---|---|---|---|---|
| 28500 | TemplateID | Y | 2 | 2 | unsigned int | Unique identifier for a MCX T7 EOBI message layout. Value: 13003 (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 MCX T7 products.<br>Belongs to the scope of Service Availability. |
| 6228 | CompletionIndicator | Y | 1 | 17 | unsigned int | Indicated whether an unit of works fitsinto a single datagram for incremental messages.<br>Value: 0 = Incomplete; 1 = Complete |
| 28841 | ApplSeqResetIndicator | Y | 1 | 18 | unsigned int | Value: 0 = Incomplete; 1 = Complete |
| 25020 | Pad5 | U | 5 | 19 | Fixed String | not used |
| 60 | TransactTime | Y | 8 | 24 | UTCTimestamp | Time in nanoseconds 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 arenot 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. |

## 4.3 Heartbeat = 13001

A functional Heartbeat message will be published regularly per product when there is no activity on the MCX T7 Enhanced Order Book Interface incremental channel. The functional Heartbeat message 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 MCX T7 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 thatwas 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 arenot transmitted.

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

## 4.4 Execution Summary = 13202

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 may be used for fast trading decisions. 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, transaction 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*, , *ExecID*, *AggressorSide*, *TradeCondition*, *LastQty*, *RestingHiddenQty* and *LastPx* respectively.
The *RestingHiddenQty* in the context of an execution (of the specified instrument) would refer to the resting hidden quantity included in the sum of *LastQty* and *RestingCxlQty*. It is set to zero, if no such quantity is involved .

| 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 MCX T7 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. |
| 2445 | Reserve3 | N | 8 | 16 | UTCTimestamp | not used |
| 5979 | Reserve1 | N | 8 | 24 | UTCTimestamp | not used |
| 17 | ExecID | Y | 8 | 32 | UTCTimestamp | Transaction time stamp |
| 32 | LastQty | Y | 8 | 40 | QuantityType | Total executed matched quantity of this match event. |
| 2446 | AggressorSide | Y | 1 | 48 | unsigned int | Value: 1 – Triggered by the buy side; 2 – Triggered by the sell side |
| 25016 | Pad1 | U | 1 | 49 | Fixed String | not used |
| 277 | TradeCondition | N | 2 | 50 | unsigned int | not used |
| 25019 | Pad4 | U | 4 | 52 | Fixed String | not used |
| 31 | LastPx | Y | 8 | 56 | PriceType | Worst price of this match. |
| 28868 | RestingHiddenQty | N | 8 | 64 | QuantityType | Quantity of executed and/or cancelled passive orders that were not displayed to the market. Set to zero, if no such quantity is involved. |
| 28869 | RestingCxlQty | Y | 8 | 72 | QuantityType | Total cancelled (deleted) quantity due to Self Match Prevention (SMP) of this 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 arenot 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<br>See also MCX T7 EOBI Schema (XSD) file. |
| 279 | MDUpdateAction | 0 | 1 | unsigned int | 0 = New<br>Type of Market Data update action. |
| 22 | SecurityIDSource | M | 1 | Fixed String | M = Marketplace<br>Marketplace assigned identifier. |

## 4.5 Order Add = 13100

An Order Add message will be published for each new order that was entered 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, in-cluding this field. |
| 28500 | TemplateID | Y | 2 | 2 | unsigned int | Unique identifier for a MCX T7 EOBI mes-sage layout. Value: 13100 (Market-DataOrder, MsgType = U21) |
| 34 | MsgSeqNum | Y | 4 | 4 | unsigned int | Message sequence number, incre-mented per product across all message types. |
| *<Message Body>* | | | | | | |
| 60 | Transaction Time | N | 8 | 8 | UTCTimestamp | Transaction timestamp in nano-seconds |
| 48 | SecurityID | Y | 8 | 16 | signed int | Unique instrument identifier. |
| *<OrderDetails>* | | | | | | |
| 21008 | Reseve2 | Y | 8 | 24 | UTCTimestamp | Not used. |
| 1138 | DisplayQty | Y | 8 | 32 | QuantityType | Quantity |
| 54 | Side | Y | 1 | 40 | unsigned int | Side of the order.<br>Value: 1 = Buy; 2 = Sell |

| Tag | Field Name | Req'd | Len | Ofs | Data Type | Description |
|---|---|---|---|---|---|---|
| 40 | OrdType | N | 1 | 41 | unsigned int | Used for cash market instruments only.<br>**1 = Market Order** Used for cashmarket instruments only.<br>Value: 1 — Description: Market Order |
| 25021 | Pad6 | U | 6 | 42 | Fixed String | not used |
| 44 | Price | N | 8 | 48 | PriceType | Price. |

**Implied Message Constants**

These constant values are to be considered as part of the above message, although they arenot 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<br>Type of Market Data update action. |
| 22 | SecurityIDSource | M | 1 | Fixed String | M = Marketplace<br>Marketplace assigned identifier. |

## 4.6 Top of Book = 13504

For derivatives market the Top of Book messages will be published via incremental and snapshot messages starting from post trading state until end of day trading state to providethe 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, in-cluding this field. |
| 28500 | TemplateID | Y | 2 | 2 | unsigned int | Unique identifier for a MCX T7 EOBI message layout. Value: 13504 (Market-DataInstrument, MsgType = U23) |
| 34 | MsgSeqNum | Y | 4 | 4 | unsigned int | Message sequence number, incre-mented per product across all message types. |
| *<Message Body>* | | | | | | |
| 60 | TransactTime | Y | 8 | 8 | UTCTimestamp | Transaction timestamp in nano-seconds |
| 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. |
| 134 | BidSize | N | 8 | 40 | QuantityType | Quantity of bid. |

| Tag | Field Name | Req'd | Len | Ofs | Data Type | Description |
|---|---|---|---|---|---|---|
| 135 | OfferSize | N | 8 | 48 | QuantityType | Quantity of offer. |
| 2449 | NumberOfBuyOrders | N | 2 | 56 | unsigned int | Number of bid orders. |
| 2450 | NumberOfSellOrders | N | 2 | 58 | unsigned int | Number of offer orders. |
| 25019 | Pad4 | U | 4 | 60 | Fixed String | not used |

**Implied Message Constants**

These constant values are to be considered as part of the above message, although they arenot 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<br>See also MCX T7 EOBI Schema (XSD) file. |
| 22 | SecurityIDSource | M | 1 | Fixed String | M = Marketplace<br>Marketplace assigned identifier. |

## 4.7 Order Modify = 13101

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 theorder will change.

| Tag | Field Name | Req'd | Len | Ofs | Data Type | Description |
|---|---|---|---|---|---|---|
| *<MessageHeader>* | | | | | | |
| 9 | BodyLen | Y | 2 | 0 | unsigned int | Number of bytes for the message, in-cluding this field. |
| 28500 | TemplateID | Y | 2 | 2 | unsigned int | Unique identifier for a MCX T7 EOBI mes-sage layout. Value: 13101 (Market-DataOrder, MsgType = U21) |
| 34 | MsgSeqNum | Y | 4 | 4 | unsigned int | Message sequence number, incre-mented per product across all message types. |
| *<Message Body>* | | | | | | |
| 60 | Transaction Time | Y | 8 | 8 | UTCTimestamp | Transaction timestamp in nano-seconds |
| 21026 | Reserve2 | Y | 8 | 16 | UTCTimestamp | Not used |
| 28855 | PrevPrice | N | 8 | 24 | PriceType | Previous order price. |
| 28867 | PrevDisplayQty | Y | 8 | 32 | QuantityType | Previous display quantity |
| 48 | SecurityID | Y | 8 | 40 | signed int | Unique instrument identifier. |
| *<OrderDetails>* | | | | | | |

| Tag | Field Name | Req'd | Len | Ofs | Data Type | Description |
|---|---|---|---|---|---|---|
| 21008 | Reserve4 | Y | 8 | 48 | UTCTimestamp | Not used. |
| 1138 | DisplayQty | Y | 8 | 56 | QuantityType | Quantity. |
| 54 | Side | Y | 1 | 64 | unsigned int | Side of the order.<br>Value: Description<br>1: Buy<br>2: Sell |
| 40 | OrdType | N | 1 | 65 | unsigned int | Used for cash market instruments on-ly.<br>**1 = Market Order** Used for cashmarket instruments only.<br>Value: Description<br>1: Market Order |
| 25021 | Pad6 | U | 6 | 66 | Fixed String | not used |
| 44 | Price | N | 8 | 72 | PriceType | Price. |

### Implied Message Constants

These constant values are to be considered as part of the above message, although they arenot 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<br>See also MCX T7 EOBI Schema (XSD) file. |
| 279 | MDUpdateAction | 1 | 1 | unsigned int | 1 = Change |
| 22 | SecurityIDSource | M | 1 | Fixed String | M = Marketplace<br>Marketplace assigned identifier. |

## 4.8 Order Modify Same Priority = 13106

An Order Modify Same Priority message will be published, if the time priority of an existing order is not changed.

| Tag | Field Name | Req'd | Len | Ofs | Data Type | Description |
|---|---|---|---|---|---|---|
| *<MessageHeader>* | | | | | | |
| 9 | BodyLen | Y | 2 | 0 | unsigned int | Number of bytes for the message, in-cluding this field. |
| 28500 | TemplateID | Y | 2 | 2 | unsigned int | Unique identifier for a MCX T7 EOBI message layout. Value: 13106 (Market-DataOrder, MsgType = U21) |

| Tag | Field Name | Req'd | Len | Ofs | Data Type | Description |
|---|---|---|---|---|---|---|
| 34 | MsgSeqNum | Y | 4 | 4 | unsigned int | Message sequence number, incre-mented per product across all message types. |
| *<Message Body>* | | | | | | |
| 21002 | Reserve1 | Y | 8 | 8 | UTCTimestamp | |
| 60 | TransactTime | Y | 8 | 16 | UTCTimestamp | Transaction timestamp in nano-seconds |
| 28867 | PrevDisplayQty | Y | 8 | 24 | QuantityType | Previous display quantity |
| 48 | SecurityID | Y | 8 | 32 | signed int | Unique instrument identifier. |
| *<OrderDetails>* | | | | | | |
| 21008 | Reserve4 | Y | 8 | 40 | UTCTimestamp | Not used |
| 1138 | DisplayQty | Y | 8 | 48 | QuantityType | Quantity. |
| 54 | Side | Y | 1 | 56 | unsigned int | Side of the order.<br>Value: 1 = Buy; 2 = Sell |
| 40 | OrdType | N | 1 | 57 | unsigned int | Used for cash market instruments on-ly. **1 = Market Order** Used for cashmarket instruments only.<br>Value: 1 = Market Order |
| 25021 | Pad6 | U | 6 | 58 | Fixed String | not used |
| 44 | Price | N | 8 | 64 | PriceType | Price. |

## Implied Message Constants

These constant values are to be considered as part of the above message, although they arenot 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 |
| 22 | SecurityIDSource | M | 1 | Fixed String | M = Marketplace Marketplace assigned identifier. |

## 4.9 Order Delete = 13102

Whenever an existing order is deleted from the order book, an Order Delete message will bepublished. The Order Delete message will contain all necessary fields needed to delete the correct order; SecurityID, 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, in-cluding this field. |
| 28500 | TemplateID | Y | 2 | 2 | unsigned int | Unique identifier for a MCX T7 EOBI message layout. Value: 13102 (Market-DataOrder, MsgType = U21) |
| 34 | MsgSeqNum | Y | 4 | 4 | unsigned int | Message sequence number, incre-mented per product across all message types. |
| *<Message Body>* | | | | | | |
| 21002 | reserve1 | N | 8 | 8 | UTCTimestamp | |
| 60 | TransactTime | Y | 8 | 16 | UTCTimestamp | Transaction timestamp in nano-seconds |
| 48 | SecurityID | Y | 8 | 24 | signed int | Unique instrument identifier. |
| *<OrderDetails>* | | | | | | |
| 21008 | Reserve2 | Y | 8 | 32 | UTCTimestamp | Not used. |
| 1138 | DisplayQty | Y | 8 | 40 | QuantityType | Quantity. |
| 54 | Side | Y | 1 | 48 | unsigned int | Side of the order.<br>Value: 1 = Buy; 2 = Sell |
| 40 | OrdType | N | 1 | 49 | unsigned int | Used for cash market instruments on-ly.<br>**1 = Market Order** Used for cashmarket instruments only.<br>Value: 1 = Market Order |
| 25021 | Pad6 | U | 6 | 50 | Fixed String | not used |
| 44 | Price | N | 8 | 56 | PriceType | Price. |

### Implied Message Constants

These constant values are to be considered as part of the above message, although they arenot 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<br>Type of Market Data update action. |
| 22 | SecurityIDSource | M | 1 | Fixed String | M = Marketplace<br>Marketplace assigned identifier. |

## 4.10 Order Mass Delete = 13103

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 tobe fully deleted.

| Tag | Field Name | Req'd | Len | Ofs | Data Type | Description |
|---|---|---|---|---|---|---|
| *<MessageHeader>* | | | | | | |
| 9 | BodyLen | Y | 2 | 0 | unsigned int | Number of bytes for the message, in-cluding this field. |
| 28500 | TemplateID | Y | 2 | 2 | unsigned int | Unique identifier for a MCX T7 EOBI mes-sage layout. Value: 13103 (Market-DataOrder, MsgType = U21) |
| 34 | MsgSeqNum | Y | 4 | 4 | unsigned int | Message sequence number, incre-mented 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 in nano-seconds |

### Implied Message Constants

These constant values are to be considered as part of the above message, although they arenot 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<br>See also MCX T7 EOBI Schema (XSD) file. |
| 279 | MDUpdateAction | 2 | 1 | unsigned int | 2 = Delete |
| 22 | SecurityIDSource | M | 1 | Fixed String | M = Marketplace<br>Marketplace assigned identifier. |

### 4.11 Partial Order Execution = 13105

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, 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 MCX T7 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.<br>Value: Description<br>1: Buy<br>2: Sell |
| 40 | OrdType | N | 1 | 9 | unsigned int | Used for cash market instruments only.<br>**1 = Market Order** Used for cashmarket instruments only.<br>Value: Description<br>1: Market Order |
| 2667 | AlgorithmicTradeIndicator | N | 1 | 10 | unsigned int | A trade is flagged as *algorithmic*, if atleast one of the matched orders wassubmitted by a trading algorithm. Ap-plicable for cash market instruments only.<br>Value: Description<br>1: Algorithmic Trade |
| 25016 | Pad1 | U | 1 | 11 | Fixed String | not used |
| 880 | TrdMatchID | Y | 4 | 12 | unsigned int | Unique identifier for each price level (match step) of a match event; it isused for public trade reporting. |

| Tag | Field Name | Req'd | Len | Ofs | Data Type | Description |
|---|---|---|---|---|---|---|
| 44 | Price | N | 8 | 16 | PriceType | The price at which the order enteredthe book. Typically it is equal to Last-Px except during auction uncrossing. |
| 60 | Transaction Time | Y | 8 | 24 | UTCTimestamp | Transaction timestamp in nano-seconds |
| 48 | SecurityID | Y | 8 | 32 | signed int | Unique instrument identifier. |
| 32 | LastQty | Y | 8 | 40 | QuantityType | 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 arenot 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 Marketplace assigned identifier. |

## 4.12 Full Order Execution = 13104

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, 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, in-cluding this field. |
| 28500 | TemplateID | Y | 2 | 2 | unsigned int | Unique identifier for a MCX T7 EOBI message layout. Value: 13104 (Market-DataOrder, MsgType = U21) |
| 34 | MsgSeqNum | Y | 4 | 4 | unsigned int | Message sequence number, incre-mented per product across all message types. |
| *<Message Body>* | | | | | | |

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Len</th><th>Ofs</th><th>Data Type</th><th>Description</th></tr>
<tr><td>54</td><td>Side</td><td>Y</td><td>1</td><td>8</td><td>unsigned int</td><td>Side of the order.<br><table><tr><td>Value</td><td>Description</td></tr><tr><td>1</td><td>Buy</td></tr><tr><td>2</td><td>Sell</td></tr></table></td></tr>
<tr><td>40</td><td>OrdType</td><td>N</td><td>1</td><td>9</td><td>unsigned int</td><td>Used for cash market instruments on-ly.<br>1 = Market Order Used for cashmarket instruments only.<br><table><tr><td>Value</td><td>Description</td></tr><tr><td>1</td><td>Market Order</td></tr></table></td></tr>
<tr><td>2667</td><td>AlgorithmicTrade-Indicator</td><td>N</td><td>1</td><td>10</td><td>unsigned int</td><td>A trade is flagged as algorithmic, if atleast one of the matched orders wassubmitted by a trading algorithm. Ap- plicable for cash market instruments only.<br><table><tr><td>Value</td><td>Description</td></tr><tr><td>1</td><td>Algorithmic Trade</td></tr></table></td></tr>
<tr><td>25016</td><td>Pad1</td><td>U</td><td>1</td><td>11</td><td>Fixed String</td><td>not used</td></tr>
<tr><td>880</td><td>TrdMatchID</td><td>Y</td><td>4</td><td>12</td><td>unsigned int</td><td>Unique identifier for each price level (match step) of a match event; it is used for public trade reporting.</td></tr>
<tr><td>44</td><td>Price</td><td>N</td><td>8</td><td>16</td><td>PriceType</td><td>The price at which the order enteredthe book. Typically it is equal to Last-Px except during auction uncrossing.</td></tr>
<tr><td>60</td><td>Transaction Time</td><td>Y</td><td>8</td><td>24</td><td>UTCTimestamp</td><td>Transaction timestamp in nano-seconds</td></tr>
<tr><td>48</td><td>SecurityID</td><td>Y</td><td>8</td><td>32</td><td>signed int</td><td>Unique instrument identifier.</td></tr>
<tr><td>32</td><td>LastQty</td><td>Y</td><td>8</td><td>40</td><td>QuantityType</td><td>Quantity executed in this fill.</td></tr>
<tr><td>31</td><td>LastPx</td><td>Y</td><td>8</td><td>48</td><td>PriceType</td><td>The price at which the order was matched.</td></tr>
</table>

### Implied Message Constants

These constant values are to be considered as part of the above message, although they arenot 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<br>Marketplace assigned identifier. |

## 4.13 Product State Change = 13300

The Product State Change message provides updates on the trading state for (all instrumentsin) 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, in-cluding this field. |
| 28500 | TemplateID | Y | 2 | 2 | unsigned int | Unique identifier for a MCX T7 EOBI message layout. Value: 13300 (Trading-SessionStatus, MsgType = h) |
| 34 | MsgSeqNum | Y | 4 | 4 | unsigned int | Message sequence number, incre-mented per product across all message types. |
| *<Message Body>* | | | | | | |
| 336 | TradingSessionID | Y | 1 | 8 | unsigned int | Product state information.<br>Value: Description<br>1: Day<br>3: Morning<br>5: Evening<br>6: After Hours<br>7: Holiday |

| Tag | Field Name | Req'd | Len | Ofs | Data Type | Description |
|---|---|---|---|---|---|---|
| 625 | TradingSessionSubID | Y | 1 | 9 | unsigned int | Product state information.<br>Value: Description<br>1: Pre Trading<br>3: Continuous<br>4: Closing<br>5: Post Trading<br>7: Quiescent |
| 340 | TradSesStatus | Y | 1 | 10 | unsigned int | Product state information.<br>Value: Description<br>1: Halted<br>2: Open<br>3: Closed |
| 2705 | MarketCondition | N | 1 | 11 | unsigned int | Indicator for stressed market conditions.<br>Value: Description<br>0: Normal<br>1: Stressed |
| 2447 | FastMarketIndicator | Y | 1 | 12 | unsigned int | Indicates if product is in state "Fast Market". This indicator refers to a product but is provided on instrument level.<br>Value: Description<br>0: No<br>1: Yes |
| 25018 | Pad3 | U | 3 | 13 | Fixed String | not used |
| 60 | TransactTime | Y | 8 | 16 | UTCTimestamp | Transaction timestamp in nano-seconds |

## Implied Message Constants

These constant values are to be considered as part of the above message, although they arenot 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 |

## 4.14 Mass Instrument State Change = 13302

The Mass Instrument State Change message provides the state information for all instrumentsof a certain instrument type or *InstrumentScopeProductComplex (1544)* within a product.
Where not all indicated instruments are affected by the new state, the exception list is populated with one entry for each such instrument.

| Tag | Field Name | Req'd | Len | Ofs | Data Type | Description |
|---|---|---|---|---|---|---|
| *<MessageHeader>* | | | | | | |
| 9 | BodyLen | Y | 2 | 0 | unsigned int | Number of bytes for the message, in-cluding this field. |
| 28500 | TemplateID | Y | 2 | 2 | unsigned int | Unique identifier for a MCX T7 EOBI mes-sage layout. Value: 13302 (Security-MassStatus, MsgType = CO) |
| 34 | MsgSeqNum | Y | 4 | 4 | unsigned int | Message sequence number, incre-mented per product across all message types. |
| *<Message Body>* | | | | | | |
| 1544 | InstrumentScope-ProductComplex | Y | 1 | 8 | unsigned int | Instrument type of affected instruments.<br>Value: Description<br>1: Simple Instrument<br>5: Futures Spread |
| 30965 | SecurityMassStatus | Y | 1 | 9 | unsigned int | The instrument status of all affected instruments.<br>Value: Description<br>1: Active<br>2: Inactive<br>4: Expired<br>9: Suspended |

| Tag | Field Name | Req'd | Len | Ofs | Data Type | Description |
|---|---|---|---|---|---|---|
| 1679 | SecurityMassTrading-Status | Y | 1 | 10 | unsigned int | The instrument trading state of all affected instruments.<br><br>Value: 2 – Trading Halt; 200 – Closed; 201 – Restricted; 202 – Book; 203 – Continuous |
| 28894 | MassMarketCondition | Y | 1 | 11 | unsigned int | Not used. This is set to 0 - Normal |
| 2447 | FastMarketIndicator | Y | 1 | 12 | unsigned int | Not used. Set to 0 by default |
| 1680 | SecurityMassTrading-Event | N | 1 | 13 | unsigned int | Identifies an event related to a *SecurityMassTradingStatus (1679)*. Used for cash market instruments only.<br><br>Value: 10 – Price volatility, auction is extended; 11 – Price volatility, auction is extended again |
| 35155 | MassSoldOutIndicator | N | 1 | 14 | unsigned int | Not used.<br>Only applicable for trading model Continuous Auction Issuer for cash market products. |
| 25016 | Pad1 | U | 1 | 15 | Fixed String | not used |
| 60 | TransactTime | Y | 8 | 16 | UTCTimestamp | Transaction timestamp in nano-seconds |

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Len</th><th>Ofs</th><th>Data Type</th><th>Description</th></tr>
<tr><td>893</td><td>LastFragment</td><td>Y</td><td>1</td><td>24</td><td>unsigned int</td><td>Indicates whether this message is thelast in a sequence of messages that to-gether convey a joint exception list of SecMassStatGrp. All messages up to the last with LastFragment = Y sharethe same root level content and anapplication first needs to combine allsingle exception lists before the MassState Change message could be ap- plied with the fully joint exception list.<br>N = Not Last Message<br>Y = Last Message<br><table><tr><td>Value</td><td>Description</td></tr><tr><td>0</td><td>N</td></tr><tr><td>1</td><td>Y</td></tr></table></td></tr>
<tr><td>146</td><td>NoRelatedSym</td><td>Y</td><td>1</td><td>25</td><td>Counter</td><td>Specifies the number of following instrument state exceptions.</td></tr>
<tr><td>25021</td><td>Pad6</td><td>U</td><td>6</td><td>26</td><td>Fixed String</td><td>not used</td></tr>
<tr><td colspan="6"><SecMassStatGrp></td><td>Variable size array, Record counter: NoRelatedSym</td></tr>
<tr><td>48</td><td>>SecurityID</td><td>Y</td><td>8</td><td>32</td><td>signed int</td><td>Unique instrument identifier.</td></tr>
<tr><td>965</td><td>>SecurityStatus</td><td>Y</td><td>1</td><td>40</td><td>unsigned int</td><td>See Instrument State Change.<br><table><tr><td>Value</td><td>Description</td></tr><tr><td>1</td><td>Active</td></tr><tr><td>2</td><td>Inactive</td></tr><tr><td>4</td><td>Expired</td></tr><tr><td>9</td><td>Suspended</td></tr></table></td></tr>
</table>

| Tag | Field Name | Req'd | Len | Ofs | Data Type | Description |
|---|---|---|---|---|---|---|
| 326 | >SecurityTrading-Status | Y | 1 | 41 | unsigned int | See *Instrument State Change*.<br><br>Value: Description<br>2: Trading Halt<br>200: Closed<br>201: Restricted<br>202: Book<br>203: Continuous |
| 2705 | >MarketCondition | Y | 1 | 42 | unsigned int | Not used. Set to 0- Normal |
| 1174 | >SecurityTrading-Event | N | 1 | 43 | unsigned int | Not used. |
| 25155 | >SoldOutIndicator | N | 1 | 44 | unsigned int | Not used. |
| 25018 | >Pad3 | U | 3 | 45 | Fixed String | Not used |

### Implied Message Constants

These constant values are to be considered as part of the above message, although they arenot transmitted.

| Tag | Field Name | Field Value | Length | Data Type | Description |
|---|---|---|---|---|---|
| 35 | MsgType | CO | 3 | Fixed String | CO = Security Mass Status |
| 22 | SecurityIDSource | M | 1 | Fixed String | M = Marketplace<br>Marketplace assigned identifier. |

## 4.15 Instrument State Change = 13301

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 MCX T7 EOBI message layout. Value: 13301 (SecurityStatus, MsgType = f) |

| Tag | Field Name | Req'd | Len | Ofs | Data Type | Description |
|---|---|---|---|---|---|---|
| 34 | MsgSeqNum | Y | 4 | 4 | unsigned int | Message sequence number, incremented per product across all messagetypes. |
| *<Message Body>* | | | | | | |
| 48 | SecurityID | Y | 8 | 8 | signed int | Unique instrument identifier. |
| 965 | SecurityStatus | Y | 1 | 16 | unsigned int | Value: 1 = Active; 2 = Inactive; 4 = Expired; 9 = Suspended |
| 326 | SecurityTradingStatus | Y | 1 | 17 | unsigned int | Value: 2 = Trading Halt; 200 = Closed; 201 = Restricted; 202 = Book; 203 = Continuous |
| 2705 | MarketCondition | Y | 1 | 18 | unsigned int | Not used |
| 2447 | FastMarketIndicator | Y | 1 | 19 | unsigned int | Not used |
| 1174 | SecurityTradingEvent | N | 1 | 20 | unsigned int | Not used |
| 25155 | SoldOutIndicator | N | 1 | 21 | unsigned int | Not used |
| 25017 | Pad2 | U | 2 | 22 | Fixed String | not used |
| 60 | TransactTime | Y | 8 | 24 | UTCTimestamp | Transaction timestamp in nano-seconds |

## Implied Message Constants

These constant values are to be considered as part of the above message, although they arenot 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<br>Marketplace assigned identifier. |

## 4.16 Product Summary = 13600

A Product Summary message will be published once each snapshot cycle, and will containattributes 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 MCX T7 EOBI message layout. Value: 13600 (Market-DataInstrument, 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 thatwas processed, regardless of message type. |
| 336 | TradingSessionID | N | 1 | 12 | unsigned int | Product state information.<br>Value: Description<br>1: Day<br>3: Morning<br>5: Evening<br>6: After Hours<br>7: Holiday |
| 625 | TradingSessionSubID | N | 1 | 13 | unsigned int | Product state information.<br>Value: Description<br>1: Pre Trading<br>3: Continuous<br>4: Closing<br>5: Post Trading<br>7: Quiescent |
| 340 | TradSesStatus | N | 1 | 14 | unsigned int | Product state information.<br>Value: Description<br>1: Halted<br>2: Open<br>3: Closed |

| Tag | Field Name | Req'd | Len | Ofs | Data Type | Description |
|---|---|---|---|---|---|---|
| 2705 | MarketCondition | N | 1 | 15 | unsigned int | Not used. |
| 2447 | FastMarketIndicator | Y | 1 | 16 | unsigned int | Not used. |
| 25022 | Pad7 | U | 7 | 17 | Fixed String | not used |

### Implied Message Constants

These constant values are to be considered as part of the above message, although they arenot 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 |

## 4.17 Instrument Summary = 13601

An Instrument Summary message will be published for each instrument in one snapshot cycle on the MCX T7 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 MCX T7 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 Trade Time |
| 68 | TotNoOrders | Y | 2 | 32 | Counter | Corresponding number of orders for this instrument. |

| Tag | Field Name | Req'd | Len | Ofs | Data Type | Description |
|---|---|---|---|---|---|---|
| 965 | SecurityStatus | Y | 1 | 34 | unsigned int | Value: Description<br>1: Active<br>2: Inactive<br>4: Expired<br>9: Suspended |
| 326 | SecurityTradingStatus | Y | 1 | 35 | unsigned int | Instrument trading state<br>Value: Description<br>2: Trading Halt<br>200: Closed<br>201: Restricted<br>202: Book<br>203: Continuous |
| 2705 | MarketCondition | Y | 1 | 36 | unsigned int | Indicator for stressed market conditions.<br>Value: Description<br>0: Normal |
| 2447 | FastMarketIndicator | Y | 1 | 37 | unsigned int | Not used |
| 1174 | SecurityTradingEvent | N | 1 | 38 | unsigned int | Not used |
| 25155 | SoldOutIndicator | N | 1 | 39 | unsigned int | Not used |
| 1227 | ProductComplex | Y | 1 | 40 | unsigned int | Value: Description<br>1: Simple Instrument<br>5: Futures Spread |
| 268 | NoMDEntries | Y | 1 | 41 | Counter | Number of entries in Market Data message for MDEntryGrp. |
| 25021 | Pad6 | U | 6 | 42 | Fixed String | not used |
| *<MDInstrumentEntryGrp>* | | | | | | Variable size array, Record counter: NoMDEntries |
| 270 | >MDEntryPx | N | 8 | 48 | PriceType | Price. |
| 271 | >MDEntrySize | N | 8 | 56 | QuantityType | Quantity. |

| Tag | Field Name | Req'd | Len | Ofs | Data Type | Description |
|---|---|---|---|---|---|---|
| 269 | >MDEntryType | Y | 1 | 64 | unsigned int | Type of market data entry.<br>Value: Description<br>2: Trade<br>4: Opening Price<br>5: Closing Price<br>7: High Price<br>8: Low Price<br>66: Trade Volume<br>101: Previous Closing Price<br>218: Open Interest<br>334: Life Time Low<br>335: Life Time High |
| 25016 | >Pad1 | U | 1 | 65 | Fixed String | Not used |
| 277 | >TradeCondition | N | 2 | 66 | unsigned int | May be set together with MDEntryType **2=Trade** or **66=Trade Volume**<br>Value: Description<br>155: Midpoint price (BB)<br>624: Trade At Close (TC) |
| 25017 | >Pad4 | U | 4 | 68 | Fixed String | not used |
| 30000 | >OILastUpdateTime | N | 8 | 72 | UTCTimestamp | This is filled when MDEntryType – 218 Open Interest. Indicates Last updated Open Interest Time |

Open Interest – Open Interest Indicates the Latest Open Interest value due to trade and Post Trade event processing

### Implied Message Constants

These constant values are to be considered as part of the above message, although they arenot 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<br>See also MCX T7 EOBI Schema (XSD) file. |
| 22 | SecurityIDSource | M | 1 | Fixed String | M = Marketplace<br>Marketplace assigned identifier. |

## 4.18 Snapshot Order = 13602

Each individual order or quote is represented as a Snapshot Order in a snapshot cycle on theMCX T7 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 MCX T7 EOBI message layout. Value: 13602 (Market-DataOrder, MsgType = U21) |
| 34 | MsgSeqNum | Y | 4 | 4 | unsigned int | Message sequence number, incremented per product across all message types. |
| *<Message Body>*<br>*<OrderDetails>* | | | | | | |
| 21008 | Reserve2 | Y | 8 | 8 | UTCTimestamp | Not used. |
| 1138 | DisplayQty | Y | 8 | 16 | QuantityType | Quantity. |
| 54 | Side | Y | 1 | 24 | unsigned int | Side of the order.<br>Value: 1 = Buy; 2 = Sell |
| 40 | OrdType | N | 1 | 25 | unsigned int | Used for cash market instruments on-ly.<br>**1 = Market Order** Used for cashmarket instruments only.<br>Value: 1 = Market Order |
| 25021 | Pad6 | U | 6 | 26 | Fixed String | not used |
| 44 | Price | N | 8 | 32 | PriceType | Price. |

### Implied Message Constants

These constant values are to be considered as part of the above message, although they arenot 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<br>See also MCX T7 EOBI Schema (XSD) file. |
| 279 | MDUpdateAction | 5 | 1 | unsigned int | 5 = Overlay |

### 4.19 Index Info = 13604

Index Info – This message is published to communicate computed price of Unique Instrument identifier for index.

| 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 MCX EOBI message layout. Value: 13604 (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. |
| 270 | MDEntryPx | Y | 8 | 16 | PriceType | Computed Index Value |
| 30000 | LastUpdateTime | N | 8 | 65 | UTCTimestamp | Indicates last updated Index value Time (in nano-seconds) |

## Implied Message Constants

These constant values are to be considered as part of the above message, although they arenot transmitted.

| Tag | Field Name | Field Value | Length | Data Type | Description |
|---|---|---|---|---|---|
| 35 | MsgType | U22 | 4 | Fixed String | Defines message type ALWAYS FIRST FIELD IN MESSAGE. (Always unencrypted) Note: A 'U' as the first character in the MsgType field (i.e. U, U2, etc) indicates that the message formatis privately defined between the sender and receiver. |

| | | | | | |
|---|---|---|---|---|---|
| 28842 | MarketDataType | 14 | 1 | unsigned int | Type of public market data, e.g., Order Book Maintenance (=1), Order Book Execution (=2), Market Segment Snapshot (=9) etc. Valid valuesare available in the MCX EOBI Schema (XSD) file. |

## 4.20 Instrument Info = 13603

Instument Info – An Instrument Info message will be published for an instrument on the MCX T7 Enhanced Order Book Interface snapshot channel whenever there is a change inthe daily price range of the instrument.

| 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 MCX EOBI message layout. Value: 13203 (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. |
| 5 | ClosePrice | N | 8 | 16 | PriceType | Close Price |
| 140 | PrevClosePrice | N | 8 | 24 | PriceType | Previous Close Price |
| 332 | UpperDailyPriceLimit | N | 8 | 32 | PriceType | Upper price Limit |
| 333 | LowerDailyPriceLimit | N | 8 | 40 | PriceType | Lower Price Limit |

## Implied Message Constants

These constant values are to be considered as part of the above message, although they arenot transmitted.

| Tag | Field Name | Field Value | Length | Data Type | Description |
|---|---|---|---|---|---|
| 35 | MsgType | U22 | 4 | Fixed String | Defines message type ALWAYS FIRST FIELD IN MESSAGE. (Always unencrypted) Note: A 'U' as the first character in the MsgType field (i.e. U, U2, etc) indicates that the message formatis privately defined between the sender and receiver. |
| 28842 | MarketDataType | 14 | 1 | unsigned int | Type of public market data, e.g., Or-der Book Maintenance (=1), Order Book Execution (=2), Market Segment Snapshot (=9) etc. Valid valuesare available in the MCX EOBI Schema (XSD) file. |