T7 Release 13.1

Market and Reference Data Interfaces

Manual

Version 1
Date 14 February 2025

Disclaimer

This publication is for informational purposes only. None of the information in this publication constitutes investment advice and does not constitute an offer to sell or a solicitation of an offer to purchase any contract, share or other financial instrument. This publication is not intended for solicitation purposes but only for use as general information. All descriptions, examples and calculations contained in this publication are for illustrative purposes only.

Deutsche Börse AG, Frankfurter Wertpapierbörse (FWB®, the Frankfurt Stock Exchange), Eurex Frankfurt AG, Eurex Deutschland and Eurex Clearing AG do not represent that the information in this publication is comprehensive, complete or accurate and exclude liability for any consequence resulting from acting upon the contents of this or another publication, in so far as no wilful violation of obligations took place or, as the case may be, no injury to life, health or body arises or claims resulting from the Product Liability Act are affected.

Securities traded on the Frankfurt Stock Exchange and Eurex derivatives (other than EURO STOXX 50® Index Futures contracts, EURO STOXX® Select Dividend 30 Index Futures contracts, STOXX® Europe 50 Index Futures contracts, STOXX® Europe 600 Index Futures contracts, STOXX® Europe Large/Mid/Small 200 Index Futures contracts, EURO STOXX® Banks Sector Futures contracts, STOXX® Europe 600 Banks/Industrial Goods & Services/Insurance/Media/Personal & Household Goods/Travel & Leisure/Utilities Futures contracts, Dow Jones Global Titans 50 IndexSM Futures contracts, DAX® Futures contracts, MDAX® Futures contracts, TecDAX® Futures contracts, SMIM® Futures contracts, SLI Swiss Leader Index® Futures contracts, RDXxt® USD - RDX Extended Index Futures contracts, Eurex inflation/commodity/weather/property and interest rate derivatives) are currently not available for offer, sale or trading in the United States nor may they be offered, sold or traded by persons to whom US tax laws apply.

The fund shares listed in XTF Exchange Traded Funds® are admitted for trading on the Frankfurt Stock Exchange. Users of this information service who legally reside outside Germany are herewith advised that sale of the fund shares listed in XTF Exchange Traded Funds may not be permitted in their country of residence. The user makes use of the information at their own risk.

Legal validity of this disclaimer

In the event that individual parts of or formulations contained in this text are not, or are no longer, legally valid (either in whole or in part), the content and validity of the remaining parts of the document are not affected.

Trademarks

The following names and designations are registered trademarks of Deutsche Börse AG or an affiliate of Deutsche Börse Group:

1585®; A7®; Buxl®; C7®; CDAX®; CEF®; CEF alpha®; CEF ultra®; CFF®; Classic All Share®; Clearstream®; CX®; D7®; DAX®; DAXglobal®; DAXplus®; DB1 Ventures®; DBIX Deutsche Börse India Index®, Deutsche Börse®; Deutsche Börse Capital Markets Partner®; Deutsche Börse Commodities®; Deutsche Börse Venture Network®; Deutsches Eigenkapitalforum®; DivDAX®; eb.rexx®; eb.rexX Jumbo Pfandbriefe®; ERS®; eTriParty®; Eurex®; Eurex Bonds®; Eurex Clearing Prisma®; Eurex Improve®; Eurex Repo®; Euro GC®; ExServes®; EXTF®; F7®; FDAX®; FWB®; GC Pooling®; GCPI®; GEX®; Global Emission Markets Access – GEMA®; HDAX®; iNAV®; L-DAX®; L-MDAX®; L-SDAX®; L-TecDAX®; M7®; MDAX®; N7®; ODAX®; ÖkoDAX®;PROPRIS®; REX®; RX REIT Index®; SCHATZ-FUTURE®; SDAX®; ShortDAX®; StatistiX®; T7®; TecDAX®; Technology All Share®; TRICE®; USD GC Pooling®; VDAX®; VDAX-NEW®; Vestima®; Xcreen®, Xemac®; Xentric®, Xetra®; Xetra-Gold®; Xpect®; Xpider®; XTF®; XTF Exchange Traded Funds®; We make markets work®

The names and trademarks listed above do not represent a complete list and, as well as all other trademarks and protected rights mentioned in this publication, are subject unreservedly to the applicable trademark law in each case and are not permitted to be used without the express permission of the registered owner. The simple fact that this publication mentions them does not imply that trademarks are not protected by the rights of third parties.

The STOXX® indices, the data included therein and the trademarks used in the index names are the intellectual property of STOXX Ltd., Zug, Switzerland and/or its licensors. Eurex' derivatives based on the STOXX indices are in no way sponsored, endorsed, sold or promoted by STOXX and its licensors and neither STOXX nor its licensors shall have any liability with respect thereto.

STOXX iSTUDIO® is a registered trademark of STOXX Ltd., Zug, Switzerland.

TRADEGATE® is a registered trademark of Tradegate AG Wertpapierhandelsbank.

EEX® is a registered trademark of European Energy Exchange AG.

Flexible is better.® is a registered trademark of Axioma, Inc.

Contents

Part I
General Overview

1 List of abbreviations

The table below shows all the abbreviations and definitions used in this document.

T7 EMDI	T7 **E**nhanced Price Level aggregated **M**arket **D**ata **I**nterface
T7 MDI	T7 netted Price Level aggregated **M**arket **D**ata **I**nterface
T7 EOBI	T7 **E**nhanced **O**rder **B**ook Market Data **I**nterface
T7 RDI	T7 Reference Data Interface
T7 RDF	T7 Reference Data File
T7 ETI	T7 Enhanced Transaction Interface
FAST	FIX Adapted for STreaming (FAST Protocol) (FAST Protocol ^{SM}). 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	Incrementals and snapshots are delivered in the same channel.
IPS	Inter-Product Spread.
Match event	Part of the matching event having a unique match price.
Out-of-band	Incrementals and snapshots are delivered on different channels.
Simple instruments	Single leg outright contracts
Complex instruments	Any combination of single leg outright contracts, e.g. Future Time Spreads.
Flexible instruments	User-defined simple instrument for TES trading.
Live-live concept	The concept whereby data is disseminated simultaneously via two separate channels called "Service A" and "Service B".
Off-book trades	Trades performed "Over the Counter".
PMAP	Presence Map
TES	T7 Entry Service
ToB	Top of Book
T7	T7 trading system developed by Deutsche Börse Group

2 Introduction

T7 offers public market and reference data via three interfaces as part of T7. All three interfaces distribute information via UDP multicast.

The T7 Market Data Interfaces are:

- The **T7 Enhanced Market Data Interface (T7 EMDI):** This interface provides *un-netted* market data. The updates of the order book are delivered for all order book changes up to a given level; all on-exchange trades are reported individually.
- The **T7 Market Data Interface (T7 MDI):** This interface provides *netted* market data. The updates of the order book are sent at regular intervals; they are not provided for every order book change and are sent significantly less frequently than the T7 EMDI. On-exchange trades are not reported individually, however statistical information (daily high/low price, last trade price and quantity) is provided instead.
- **T7 Enhanced Order Book Interface (T7 EOBI):** This interface provides the entire visible order book, by publishing information on each individual order and quote along with state information in *un-netted* manner. All on-exchange trades are reported individually. See “T7 Enhanced Order Book Interface - Manual ”.

The T7 EMDI and T7 MDI provide the following information to the participants:

- Price level aggregated order book depth and trade statistics.[1]
- Product and instrument states.
- Quote requests and cross requests.
- Information on newly created complex and flexible instruments (derivatives only)

Reference data is sent separately per market by:

- The **T7 Reference Data Interface (T7 RDI)**: This interface provides reference data for products and instruments that are available for trading on T7. The reference data is delivered on a product and instrument level. Every tradable object is referenced by a unique identifier, for this reason the reference data information is absolutely essential for any trading application.
- The **T7 Reference Data File (T7 RDF)**: Reference data is delivered as a start of day file and as regularly updated[2] intraday files .

T7 EMDI, T7 MDI and T7 RDI publish market and reference data information following FIX 5.0 SP2 semantics and are FAST 1.2[3] encoded. If any messages are lost, complete recovery is possible because every message is published on two identical services (A and B) with different multicast addresses (live-live concept). In the unlikely case that a message is lost on both services, participants can take advantage of the respective snapshot messages and rebuild the order book.

The scope of this manual is T7 EMDI, T7 MDI, T7 RDI and T7 RDF. For details regarding T7 Enhanced Order Book Interface (T7 EOBI), please see “T7 Enhanced Order Book Interface - Manual ”.

T7 EMDI, T7 MDI, T7 RDI and T7 RDF do not offer any layout-level backward compatibility feature between two releases, and within the lifetime of a release T7 reserves the right to change the behavior of some fields in the different layouts.

[1] Eurex Off-Book trades replay, settlement prices and open interest information are provided by the T7 Extended Market Data Service.
[2] Currently with an update interval of 5 minutes.
[3] FAST 1.1 templates are provided as well.

2.1 Purpose of this document

The purpose of this document is to provide guidance for programmers during development of applications that read the *T7 Market & Reference Data Interfaces*.

It covers a complete reference for the three multicast based public interfaces[4], describes the general business behaviour and provides concepts for the implementation.

The most recent version is available at:

www.eurex.com > Support > Initiatives & Releases > T7 Release 13.1 > System Documentation > Market & Reference Data Interfaces

or

www.xetra.com > Technology > T7 trading architecture > System documentation > Release 13.1 > Market and Reference Data Interfaces.

2.2 Main audience

The target audience of this interface specification is experienced software developer support staff that may be involved in development/support activities for the *T7 Market & Reference Data Interfaces*.

Prior knowledge of developing for cash or derivative markets is beneficial but not a prerequisite. Knowledge in a programming language is expected. Programmers who have no experience in a market data interface environment can gain a basic understanding of the feed behaviour by reading Part II (How to guide). This manual does not attempt to cover basic knowledge of programming techniques and software development.

2.3 Data feeds

All interfaces deliver public reference and market data in the form of snapshots and incrementals as can be seen in Figure 1. The two public market data interfaces, the **T7 EMDI** for a high bandwidth network and the **T7 MDI** for a low bandwidth network, disseminate information across the T7 network to the receiving application. The **T7 RDI** is considered for participants with a high bandwidth network while the **T7 RDF** should be used if only a low bandwidth network is available.

[4] T7 EMDI, T7 MDI and T7 RDI. T7 EOBI is covered in a separate document.

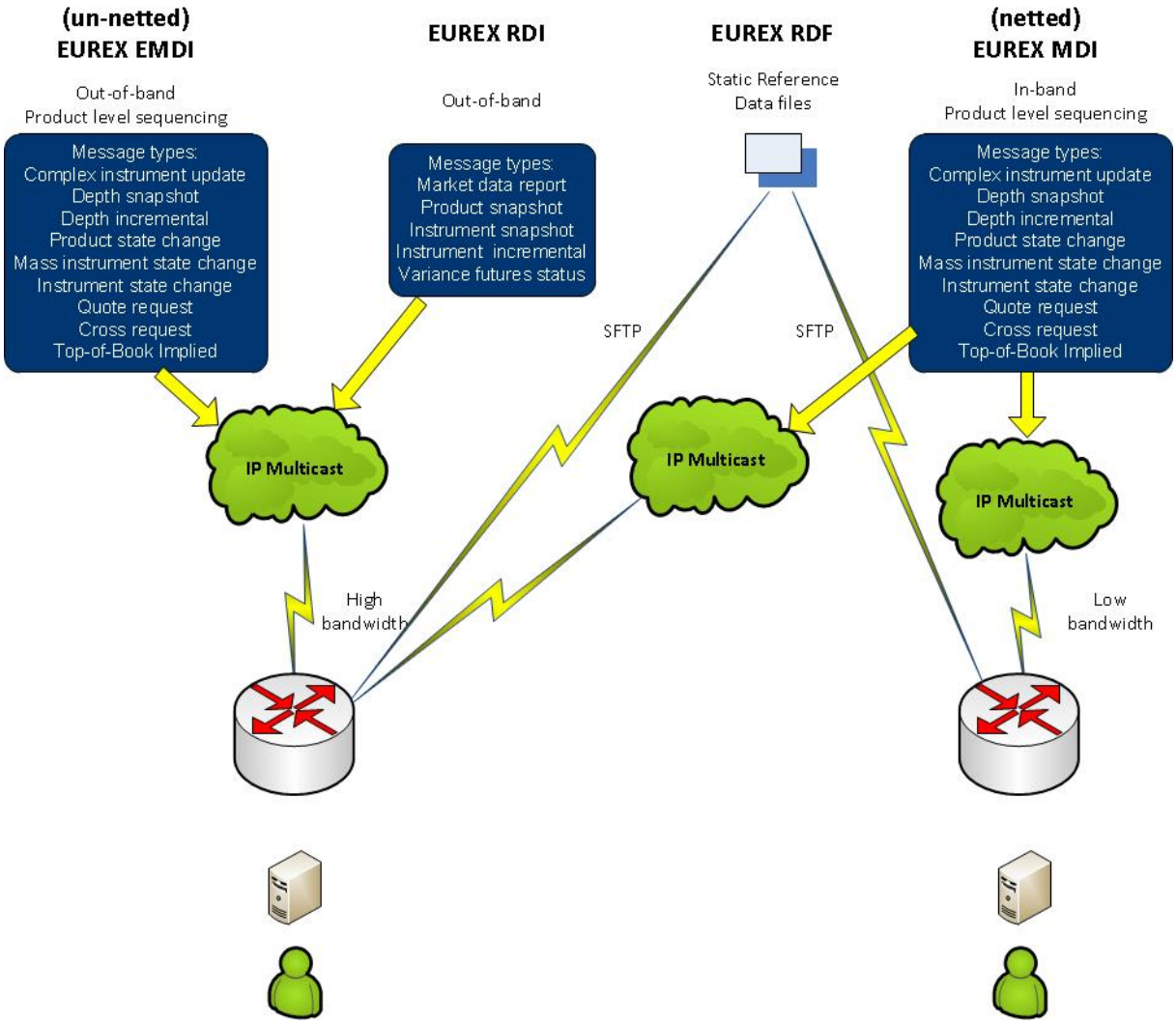

Figure 1: *For example Eurex reference data and Market data interfaces*

2.3.1 Reference data interface

Public reference data delivered by T7 RDI contains the technical configuration, e.g. multicast address and port combinations for both market data interfaces for all products and instruments.

There are separate reference data feeds per marketID.

Multicast addresses and port information do not change during trading hours.

The **reference data snapshot feed** contains two message types: Constant number of snapshots and a variable number of incrementals.

The **reference data incremental feed** delivers reference information about complex and flexible instruments. For cash market products it delivers reference information about the activation and attribute change of instruments setup as covers.

2.3.2 Market data interfaces

The T7 EMDI and the T7 MDI disseminate public market data information in the form of incrementals (event driven) and snapshots (time driven).

The **market data snapshot feed** can be used to recover lost market data or build up the current order book. Receiving applications are not expected to be permanently subscribed to this feed.

The **market data incremental feed** should be subscribed throughout the trading day for receiving order book updates. All incoming messages should be applied to the copy of the order book maintained by the member applications in order to have the latest information.

2.4 Interface version number

Each of the interfaces described in this manual has a version number. The version numbers are listed at the beginning of the FAST XML template files.

This manual relates to the following interface version numbers:

- T7 EMDI: 131.290.0
- T7 MDI: 131.290.0
- T7 RDI: 131.290.0

2.5 Deutsche Börse customer support

T7 support is available 24hrs on business days and may be contacted as follows:

	T7 Contact List	
Technical Support (global)	VIP number	cts@deutsche-boerse.com
Technical Support (USA)	VIP number	cts@deutsche-boerse.com
Service Portal	Create problem ticket	www.eurex.com > Support > Member Section
Eurex Functional Helpdesk (Equity/Index)	+49-69 2111 1210	eurextrading@eurex.com
Eurex Functional Helpdesk (Fixed Income)	+49-69 2111 1240	eurextrading@eurex.com
Market Supervision (Xetra)	+49-69-2111 1400	cmmarketcontrol@deutsche-boerse.com
Market Supervision (Frankfurt)	+49-69-2111 1400	cmmarketcontrol@deutsche-boerse.com

Table 1: *T7 contact list*

2.6 Further reading matter for this topic

This document is designed as an independent learning and reference manual. However, for background information related to network connectivity, FAST/FIX messages or trading related information (functional), further documents are recommended.

The documents listed below provide useful information.

FAST- and FIX-related documents:

- **FAST specification documents**: Explains all FAST rules in detail. FAST 1.2 is the summary of the FAST 1.1 specification plus the extension Proposal.
  FIX Adapted for Streaming (FAST)
- **FIX specification documents**: FIX-messages and FIX-tags
  FIX Standards
- **FIX-Tags**: Specifies all FIX-Tags
  FIXimate

T7 related documents:

www.eurex.com > Support > Initiatives & Releases > T7 Release 13.1

or

www.xetra.com > Technology > T7 trading architecture > System documentation > Release 13.1

- **T7 Functional and Interface Overview**: This document provides an overview of the T7 trading architecture. It describes the major functional and system changes, and provides a high level description of the interface landscape.
- **T7 Functional Reference**: Provides a detailed description of the business functionality that is available in T7.
- **T7 Extended Market Data Service Manual**: This document provides the information about the add-on services for market and reference data, e.g. intraday settlement prices, open interest, etc...
- **T7 Enhanced Order Book Interface Manual**: This manual describes the concepts and the messages used by this interface.
- **T7 Enhanced Trading Interface Manual**: It contains a detailed description of the concepts and messages used by this trading interface.

2.7 How to read this document

This manual covers the T7 EMDI and T7 MDI as well as the T7 RDI. Differences in functionality between the T7 EMDI and the T7 MDI are described in separate sub sections.

For example, section 7.4.2, Recovery (T7 MDI), on page 41 refers to the “netted” T7 MDI only. Participants who are interested in the “un-netted” T7 EMDI can ignore this sub chapter. This document consists of three parts:

- **Part I** (General Overview) introduces the interface for beginners.
- **Part II** (How to guide) provides methods and hands-on guidance.
- **Part III** (Reference) is a comprehensive reference with details on various message layouts in table format. A typical table would be the following:

Delivered on: reference data snapshot feed

Tag	Field Name	Req'd	Data Type	Description
35	MsgType	Y	string	User defined message **Value** / **Description**: 0 – Beacon
<GroupName > (optional) group starts				
<SequenceName > sequence starts				
...	...	...	...	...
...	...	...	...	...
<SequenceName > sequence ends				
<GroupName > (optional) group ends				

Table 2: *Typical FIX message description*

Interpreting the fields above:

- **Delivered on**: Specifies the feed which delivers the specific message. A message can be delivered on more than one feed.
- **Tag**: Describes the FIX Tags
- **Field Name**: Describes the FIX-name.
- **Req'd**: Describes whether or not the field is included within the message after FAST-decoding, purely from the FIX-point of view. This does not refer to a FAST-rule, e.g. operators or Presence Map (PMAP) in FAST.
- **Data Type**: FAST data type. This information is also provided in the XML FAST templates.
- **Description**: This column contains an explanation of the FIX-field and it's "valid values" in table format for this particular message.
- **GroupName, SequenceName**: The names correspond with the groups and sequences defined in the FAST XML templates.

Cross references to other chapters within this document and the glossary are provided in blue color.
Example: More information is provided in section 9.1, Reference data messages.

In this document, the terms "incrementals" and "snapshots" are used in various contexts. Within this document "incrementals" and "snapshots" refer either to messages of the market data feed or to messages of the reference data feed. The actual meaning can be inferred from the context.

Note: Important statements made in this manual are highlighted with a shadow box.

3 Differences between the interfaces

A feed is a message flow of logically grouped messages, e.g. the *Depth incremental* and *Product state change* messages for a particular product are grouped together within the incremental feed of T7 EMDI. The following diagram illustrates the available feeds for the three multicast based public interfaces:

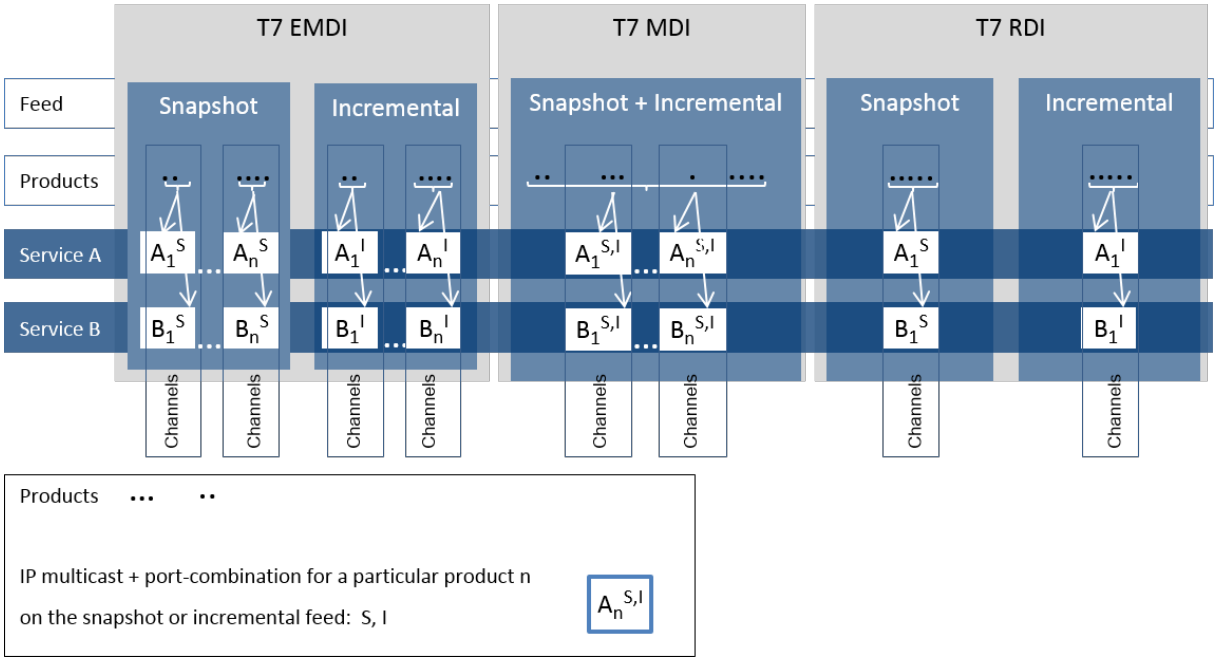

Figure 2: *Overview of the three interfaces*

The T7 RDI is published on exactly one snapshot channel, indicated by (A_1^S) and one incremental channel (A_1^I). The T7 EMDI has multiple channels that have either snapshots (A_1^S) to (A_n^S) and multiple incremental channels (A_1^I) to (A_n^I). The T7 MDI has the snapshots and incrementals combined over multiple channels ($A_1^{S,I}$) to ($A_n^{S,I}$).

The snapshot and incremental messages for the **T7 EMDI** are delivered via separate feeds (out-of band) and need to be synchronized. Each feed consists of several channels, each of which delivers the information for a group of products.

Several partitions, each with a unique *SenderCompID (49)*, may contribute to the same multicast address as shown in figure 16 on pg. 80. The *SenderCompID (49)* is unique across all partitions. However, it should not be relied upon as under unlikely but possible conditions on the exchange this is not true.

In contrast to the T7 EMDI, the snapshot and incremental messages for the **T7 MDI** are sent on one feed only (in-band), therefore there is no need to synchronize both messages. The feed is also divided into several channels grouped on product basis.

The snapshot and incremental feed for the **T7 RDI** are delivered via separate channels (out-of band) and need to be synchronized. In contrast to the order book information, the snapshot and incremental feeds are not divided into further channels.

All feeds are sent on two different multicast addresses via different physical connections (Service A and B). Service A and Service B are identical in terms of the information provided, i.e. the packet contents, sequence numbers and sequence in which packets are sent is the same. This is called “live-live” concept.

Product groups are distributed across several partitions on the T7 backend side. Service A and Service B cannot be published at exactly the same time.

3.1 Distribution sequence for T7 EMDI

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* (5948). It is available in the *Product snapshot* message and in the *packet header* and contains the number of the partition for the product of interest. The *PartitionID* (5948) never changes intraday.

Example: A PartitionID = 8 indicates an even partition and therefore Service A is published before Service B.

Current production data indicate an average time difference of about 10 - 15 µs; the cable length for both Service A and Service B within the co-location is the same, i.e. both services have the same propagation delay.

The multicast addresses for both of these services are disseminated in the product reference information. Due to the inherently unreliable nature of the UDP protocol, data packets may be lost in the transmission network. Therefore members are advised to join both services to reduce the probability of data loss.

3.2 Distribution sequence for T7 MDI / T7 RDI

The rule for the **distribution sequence** across partitions is as follows:

Even and odd partitions: Publish on Service A first, then on Service B.

Example: The *PartitionID* (5948) for T7 MDI and T7 RDI is not available in the *Packet header* but in the *Product snapshot* message. However, the *PartitionID* (5948) doesn't need to be considered because Service A is always published first regardless of the partition.

3.3 Choosing between T7 EMDI and the T7 MDI

Both types of interface, un-netted and netted, provide market information via multicast using a price-level aggregated order book (as opposed to, for example, order-by-order feeds) but they have different bandwidth requirements and service levels.

- The **T7 Enhanced Market Data Interface (un-netted)** disseminates every order book change up to the configured depth and all on-exchange trades without netting. This interface is designed for participants that rely on *low-latency* order book updates and data completeness. The un-netted market data is partitioned over several channels; each channel provides information about a group of similar products. As the market becomes busier, the number of messages (and therefore bandwidth usage) increases.
- The **T7 Market Data Interface (netted)** has a *lower bandwidth* requirement compared to the un-netted version. This interface is designed for participants who do not need to see every order book update, this has the advantage of keeping the infrastructure costs low. Snapshot and incremental updates are sent via the same IP multicast address and port combination. The order book depth may be lower than for EMDI.

  This interface aggregates the order book changes over a specified time interval, which is published in the *Product snapshot* message via field *MarketDepthTimeInterval (2563)*. The intervals of the incremental messages are often higher than MarketDepthTimeInterval. This is because a smart bandwidth management logic considers the actual overall bandwidth consumption. This interface has less price levels than the T7 EMDI. Furthermore, only statistical information is provided for on-exchange trades as well as the price and quantity of the last on-exchange trade in the netting interval.

The following table shows the main differences between the T7 EMDI and the T7 MDI:

Area	T7 EMDI	T7 MDI
In-band/Out-of-band delivery	Incrementals and snapshots are delivered via different channels, i.e. out-of-band delivery. *LastMsgSeqNumProcessed* in the snapshot feed provides a link between incremental and snapshot feed, as it carries the sequence number of the last message sent on the incremental feed. Snapshots are needed only for start-up/recovery.	Incrementals and snapshots are delivered on the same channel, i.e. in-band delivery. Snapshots might contain new information. A flag (*RefreshIndicator*) within the snapshot indicates whether it has to be applied or not. *LastMsgSeqNumProcessed* is not used.
Sequence numbers on message level	Messages on the market data incremental feed have their own sequence number range per product; *MsgSeqNum's* exist on the depth incremental feed only as shown in table 14 on pg. 39.	Messages on the combined market data incrementals + snapshot feed have one sequence number range per product as shown in table 15 on pg. 41.
Trade Volume Reporting	Trade Volume Reporting is provided. Each on-exchange trade is reported individually.	Only statistical information (daily high/low price and total traded quantity) and last trade information is provided.

Area	T7 EMDI	T7 MDI
Packet header	A Performance Indicator[5] is provided for incrementals within the Packet Header as shown in figure 17 on pg. 83.	A Performance Indicator does not exist as shown in figure 18 on pg. 84.
Functional beacon message	A *functional Beacon* message on a product level including the last valid *MsgSeqNum* is sent if no other message has been sent for a configured time period.	Snapshots act as *functional Beacon* message, hence no separate *functional Beacon* messages are provided.

Table 3: *Main differences between the T7 EMDI and the T7 MDI*

Both interfaces, un-netted and netted, provide different recovery time intervals to offer the participants the opportunity to implement a suitable public market data recovery mechanism. The recovery time interval, *MDRecoveryTimeInterval(2565)*, of a product for each interface is available in *Product snapshot* message via the T7 RDI, as well as in file format via the T7 RDF.

3.4 Choosing between T7 RDI and T7 RDF

Reference data is provided via the **T7 RDI** and in file form as compressed **Reference Data Files (RDF)** in FIXML-layout, updated approximately every 5 minutes via the Common Report Engine [6] (CRE, only for derivatives markets).

The initial reference data file generated at the start of day contains the “reference data snapshots” available from the previous day without complex (and scaled simple) instruments. During the actual trading, multiple incremental files are created as complex and flexible instruments are added. Complex (and scaled simple) instruments from the previous day and new complex instruments predefined by the exchange are sent in incremental files before the start of the actual trading.

Please note that the intraday changes to reference data are also published in form of the *Complex instrument update* and *Flexible instrument update* messages via the market data incremental feed of the T7 EMDI and the market data feed of the T7 MDI. During normal operations participants do not need to listen to the incremental feed of the **T7 RDI**, because the *Complex instrument update* and *Flexible instrument update* messages can be received on the market data feed as well. Furthermore, market data for new complex and flexible instruments is never provided ahead of their reference data on EMDI or MDI but may come ahead of its publication via RDI.

Participants have the choice between the two different reference data sources. However, it is assumed that bandwidth conscious users will use **T7 RDF** for the start of day processing and intraday re-starts. The reference data file is provided once the system is available.

The following table shows the main differences between the T7 RDI (message based) and the T7 RDF (file based):

[5] Time between arrival of an incoming order/quote transaction on the T7 gateway and send time of the corresponding outgoing market data

[6] For more information please see the "Common Report Engine User Guide"

Area	T7 RDI: Message based	T7 RDF: File based
Reference Data	High bandwidth users can use the multicast based Reference Data Interface.	Low bandwidth users can use Start-Of-Day Reference Data Files and apply each Intra-day Reference Data File as they become available. A late starting application can always retrieve the latest picture of the reference data by this method.

Table 4: *Differences between the T7 RDI and the T7 RDF*

It is important to note that the T7 reference data interface, T7 RDI and T7 RDF, is intended to be available prior to the daily T7 start up processing. This service will normally be available during non-trading days to support participants with the T7 public reference data. However, during any kind of infrastructure maintenance this service will not be available on non-trading days.

4 Overview of the T7 Public Interfaces

This chapter describes the public market data provided by the market and reference data interfaces.

4.1 Infrastructure requirements

The T7 market and reference data interfaces disseminate market and reference data over the T7 multicast network. A router which is capable of handling IP multicast is required for accessing this interface. The multicast management protocol is IGMPv2. When utilizing IGMPv3, the IGMPv2 compatibility mode must be enabled.

4.2 Trading states

State changes are disseminated over both the T7 EMDI and the T7 MDI market data feeds. Trading state information is not communicated over the T7 Enhanced Transaction Interface (T7 ETI) or FIX interface.

The T7 EMDI and the T7 MDI market data feeds follow the FIX protocol for the publication of trading state information. The T7 product and instrument states are displayed by these interfaces as shown in the following tables.

Section 9.11, Trading states for a sample business day for derivates illustrates state messages for a typical business day. The hours of operations for the T7 system is provided in Section 4.8, Hours of operation/availability of messages.

4.2.1 Product State Changes

The product state is published with a *product state change* message (FIX *TradingSessionStatus*, MsgType = h). In this message, the product state can normally be found in the field *TradingSessionSubID (625)*. Only for quiescent product states, the field *TradingSessionID (336)* must be evaluated additionally to determine the actual product state.

product state change message			
T7 Product State	**FIX TradingSessionID (336)**	**FIX TradingSessionSubID (625)**	**FIX TradeSesStatus (340)**
Start of Day	3 = Morning	7 = Quiescent	3 = Closed
Pre-Trading	3 = Morning	1 = Pre-Trading	2 = Open
Trading	1 = Day	3 = Continuous	2 = Open
Closing	1 = Day	4 = Closing	2 = Open
Post-Trading	5 = Evening	5 = Post-Trading	2 = Open
End of Day	5 = Evening	7 = Quiescent	3 = Closed
Post End of Day	6 = After-Hours	7 = Quiescent	3 = Closed
Halt	1 = Day	7 = Quiescent	1 = Halted
Holiday	7 = Holiday	7 = Quiescent	3 = Closed

Table 5: *Product states*

A Halt state is additionally indicated by the FIX field *TradSesStatus (340)* containing the value 1 = Halted. A Fast Market is reported with the same message type using the new FIX field *FastMarketIndicator (2447)* which can take the values 0 = No or 1 = Yes.

A Stressed Market is reported with the same message type using the new FIX field *MarketCondition (2705)* which can take the values 0 = Normal or 1 = Stressed. For cash products MarketCondition is only set as part of instrument state.

The product TES activity status is independent of the on-exchange product state. TES activity status is reported using the new FIX field *TESTradSesStatus (25044)*.

product state change message	
TES Activity Status	**FIX TESTradSesStatus (25044)**
Off	3 = Closed
On	2 = Open
Ended	5 = Pre-Close
Halted	1 = Halted

Table 6: *TES Activity Status*

4.2.2 Instrument State Changes

The instrument state is published with an *instrument state change message* (FIX *SecurityStatus*, MsgType = f) in case of a single instrument, or with a (FIX *SecurityMassStatus*, MsgType = CO) message in case that all or most of the instruments of a product and of a specific instrument type[7] change their state.

- In the *instrument state change* message (FIX *SecurityStatus*, MsgType = f), the instrument state can be found directly in the field *SecurityTradingStatus* (326).
- In the *mass instrument state change* message (FIX *SecurityMassStatus*, MsgType = CO), the instrument state can be found in the field *SecurityMassTradingStatus* (1679). This message may contain an exception list of instruments that have a different instrument state. The exception list contains the instrument state in the field *SecurityTradingStatus* (326) for each of these instruments.

[7] Instrument types distinguish simple instruments (option series, futures contracts) and various types of complex instruments

(mass-) instrument state change message	
T7 Instrument State	**FIX *SecurityTradingStatus (326)* /** **FIX *SecurityMassTradingStatus (1679)*** 2 = Trading Halt, 212 = IPO Auction, 213 = IPO Auction Freeze, 216 = Freeze and 217 = Trade At Close are applicable for cash market instruments only.
Trading Halt	2 = Trading Halt
Closed	200 = Closed
Restricted	201 = Restricted
Book	202 = Book
Continuous	203 = Continuous
Opening Auction	204 = Opening Auction
Opening Auction Freeze	205 = Opening Auction Freeze
Intraday Auction	206 = Intraday Auction
Intraday Auction Freeze	207 = Intraday Auction Freeze
Volatility Interrupt Auction	208 = Circuit Breaker Auction
Volatility Interrupt Auction Freeze	209 = Circuit Breaker Auction Freeze
Closing Auction	210 = Closing Auction
Closing Auction Freeze	211 = Closing Auction Freeze
IPO Auction	212 = IPO Auction
IPO Auction Freeze	213 = IPO Auction Freeze
Pre Call	214 = Pre Call
Call	215 = Call
Freeze	216 = Freeze
Trade At Close	217 = Trade At Close
Volatility Interrupt Auction Triggered By Static Limit Breach	220 = Circuit Breaker Auction Triggered By Static Limit Breach
Volatility Interrupt Auction Triggered By Static Limit Breach Freeze	221 = Circuit Breaker Auction Triggered By Static Limit Breach Freeze

Table 7: *Instrument states*

The field *FastMarketIndicator (2447)* is also contained in the *mass instrument state change* message; each instrument state message also contains the information about whether the product that the instrument belongs to is in a Fast Market state. This implies that a *mass instrument state change* message is sent when a product is set to Fast Market (or back) without a change in the instrument states.

The status of the instrument (as opposed to the instrument state) distinguishes active and published instruments and is contained in the field *SecurityStatus* (965).

A Stressed Market is reported with new FIX field *MarketCondition (2705)* which can take the values 0 = Normal or 1 = Stressed.

4.3 Overview of the various message types

The various message types can be divided into "Service Messages" and "Data Messages".

4.3.1 T7 RDI

Service messages:

- *Technical heartbeat message* see 11.1.5.
- *Functional beacon message* see 11.1.4 contains the last valid *MsgSeqNum* and is only sent on the reference data incremental feed when there is no activity for a certain amount of time. No functional beacons are sent on reference data snapshot feeds.

Data messages:

- *Market data report message* flags the start and end of reference data. Each message is flagged by a start/stop event identifier.
- *Product snapshot message* contains product specific reference data.
- *Instrument snapshot message* contains a snapshot of instrument specific reference data.
- *Instrument incremental message* used for complex and flexible instruments. After partition startup identical messages are also sent on the market data incremental feed of the T7 EMDI as well as on the market data feed of the T7 MDI. In cash markets *Instrument incremental message*s are used to report the activation and attribute change of instruments setup as covers. Here no additional messages are sent on the market data incremental feed of the T7 EMDI as well as on the market data feed of the T7 MDI.
- *Variance futures status message* used to convey information specific to variance future instruments either at the start of day or intra-day.
- *Total return futures status message* used to convey information specific to total return future instruments either at the start of day or intra-day.
- *Trade At Reference Price status message* used to convey information specific to trade at reference price instruments either at the start of day or intra-day.

4.3.2 T7 EMDI/MDI

Service messages:

- *Technical heartbeat message* see 11.1.5.
- *Functional beacon message* see 11.1.4 contains the last valid *MsgSeqNum* of each product and is only sent on the market data incremental feed when there is no activity in a product for a certain amount of time. No functional beacons are sent for T7EMDI snapshot and T7 MDI feeds.

Data messages:

- *Depth snapshot message* is used to send a snapshot of all price levels of the order book and statistical information about on-exchange trades. This message can be used whenever the order book needs to be rebuilt.
- *Depth incremental message* is used to receive updates on the initial order book.
- *Top Of Book Implied message* is used to send top of book information resulting from synthetic Inter Product Spread (IPS) matching opportunities.
- *Product state change message* is used to publish the state of the T7 products.

- *Mass instrument state change message* provides the state information for all instruments of a product. This message can publish different states for instruments of the same product, e.g. in case of a volatility interruption the front month could be in a different state than the back month.
- *Instrument state change message* provides state information for a single instrument.
- *Quote request message* provides requests to market makers to enter quotes for specified instruments.
- *Cross request message* is sent once a participant announces the intention to enter a cross trade.
- *Complex instrument update message* is used to publish complex instruments. This message is sent via the market data incremental feed of the T7 EMDI and the market data feed of the T7 MDI. A message is sent for each complex instrument.
- *Flexible instrument update message* is used to publish flexible instruments. This message is sent via the market data incremental feed of the T7 EMDI and the market data feed of the T7 MDI. A message is sent for each newly created flexible instrument.

A detailed description of the message types listed above is given in section 11, Detailed data feed description and layout.

4.4 What is not included in these interfaces

The following information is **not** provided:

- For auctions in derivative products, the best bid/ask prices are disseminated at price level 1 **without** a quantity. If a potential auction price is calculated, it is also sent without the quantity. Order book depths are **not** delivered during auctions, only top of book information is disseminated. For auctions in cash market market products the dissemination depends on the specific product setting.
- Market Supervision News is **not** provided. This information is available via the T7 ETI in recoverable form.
- Prices for external underlyings are **not** provided.
- Retransmission functionality is **not** provided, but recovery is possible from the respective other service (A or B). In case a message is lost a snapshot can be used to rebuild the order book.
- Implied prices are only sent for Best Market, they are **not** sent for the order book depth except for top of book.

4.5 FIX over FAST

FIX messages are sent out in FAST 1.2 encoded format. The receiving software decodes the FAST messages according to the FAST 1.2 rules.

Note: FAST 1.2 templates and FAST 1.1 compatible templates are provided.

After the decoding process, the actual FIX message can be built by applying the FIX structure to the decoded message. The detailed process is shown in Part II, FIX/FAST-Implementation.

Participants need a standard FAST template based decoder in order to be able to use the T7 EMDI, T7 MDI and T7 RDI. Alternatively participants can use their own FAST decoder implementation.

4.6 Freedom of choice

T7 does not need to provide any software for accessing the services offered. The T7 market and reference data interfaces can be accessed using any platform capable of receiving multicast data feeds.
Participants can use any operating system, compiler version or programming language in order to develop or use specific third party applications that are tailored to their requirements.

4.7 Testing

It is recommended to test the functionality application logic sufficiently in a simulation environment.

Receiving applications must be able to cope appropriately with a variety of T7 service fail-over scenarios. For this purpose, special test scenarios are offered in a simulation environment.

4.8 Hours of operation/availability of messages

- The product state "*Pre-Trading*" (*TradingSessionSubID (625)* = 1) e.g. for many of the EUREX products begins at **7:30 CET**.
- "*Post-Trading*" (*TradingSessionSubID (625)* = 5) e.g. for some Eurex products lasts until **22:30 CET**.

For detailed information on trading hours please refer to:

www.eurex.com > Trade > Trading hours.

- T7 is available from approximately **01:00-01:10 CET**. For specific benchmark futures products it is recommended to start applications already before **1:00 CET**.
- Market data messages are sent from the time a product changes to the state "Start-Of-Day" and stops while in state "Post-End-Of-Day". During that period *depth snapshots* are sent.

  The reference data is independent to any one product state so it has its own schedule.
- Receiving applications are expected to stay connected from product state "Start-Of-Day" until product state "End-Of-Day".

The following table provides further details about the availability of messages per instrument state:

State	Market Data Orderbook	Market Data State Info
Continuous	Yes	Yes
Auction	Yes	Yes
Freeze	Yes	Yes
Book	No	Yes
Restricted	No	Yes
Closed	No	Yes

Table 8: *Availability of messages per instrument state*

Part II
How to guide

5 FIX/FAST-Implementation

This chapter describes the message structure for the three interfaces. It also provides the basic FAST-rules used by the interface and describes the basic steps from receiving a FAST datagram, decoding it and building FIX-messages out of it.

The FAST 1.2 specification is provided as an extension to the FAST 1.1 specification. The documents can be found under the following links:

FAST Specification (Version 1.1), FAST version 1.2 Extension Proposal
FIX Adapted for Streaming (FAST)

5.1 Structure of Messages

The three public interfaces disseminate data in UDP datagrams in network byte order also known as big endian byte order. This includes vector encoded numbers. A UDP datagram has the following structure:

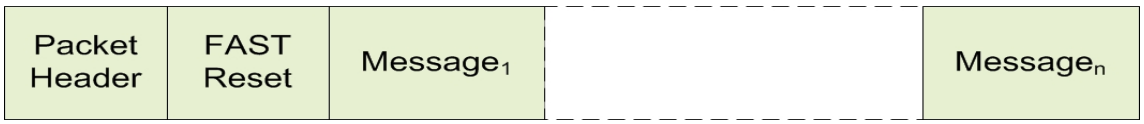

Figure 3: *Structure of a UDP datagram*

- The UDP datagram starts with the packet header message as shown in section 11.1.2.
- Followed by a FAST reset message.
- Followed by the actual message ($Message_1$).
- Possibly followed by one or more messages ($Message_2$ - $Message_n$).

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

- PMAP (Presence Map).
- TID (Template ID).
- Data Part.

This is shown in the following diagram:

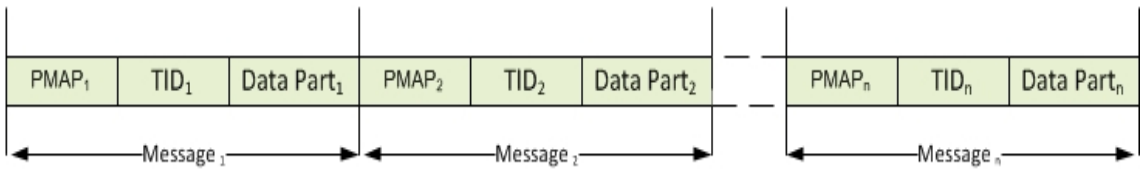

Figure 4: *Structure of consecutive messages within one datagram*

One UDP datagram contains one or more FAST encoded FIX 5.0 SP2 messages. The UDP protocol adds a 28 byte header to every packet (20 byte IP header plus 8 byte UDP protocol header). Due to the unreliable nature of UDP, every UDP datagram is self contained; there is no dependency across datagrams.

5.2 FAST terminology

5.2.1 FAST reset message

The T7 Market Data Interfaces use **global dictionary** scope for FAST operators[8]. All operators share the same dictionary regardless of the template and application type. The *FAST reset* message is inserted at the start of every datagram to explicitly reset all the dictionaries.

5.2.2 Presence Map (PMAP)

The presence map is a bit combination indicating the presence or absence of a field in the message body, one bit in the PMAP for each field that uses a PMAP bit according to the FAST type. The allocation of a bit for a field in the presence map is governed by the FAST field encoding rules.

5.2.3 Template ID (TID)

The template identifier is represented by an integer number and points to a specific FAST template which describes the layout and characteristics of the message to be decoded. The FAST XML files are provided in section 13, FAST templates.

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

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

The following list contains the message types and their corresponding template identifiers used with the three T7 interfaces:

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

Message	TID T7 RDI	TID T7 EMDI	TID T7 MDI
Functional Beacon (aka Functional Heartbeat)	109	109	-
Packet header for T7 RDI / EMDI / MDI	74	62	67
FAST Reset Message	120	120	120
MarketDataReport	125	-	-
ProductSnapshot	122	-	-
InstrumentSnapshot	123	-	-
InstrumentIncremental	121	-	-
VarianceFuturesStatus	162	-	-
TotalReturnFuturesStatus	514	-	-
TradeAtReferencePriceStatus	515	-	-
ComplexInstrumentUpdate	-	124	105
DepthSnapshot	-	93	101
DepthIncremental	-	94	102
QuoteRequest	-	95	106
CrossRequest	-	96	107
ProductStateChange	-	97	108
MassInstrumentStateChange	-	99	104
InstrumentStateChange	-	98	103
TopOfBookImplied	-	512	513
FlexibleInstrumentUpdate	-	100	110
ScaledSimpleInstrumentUpdate	-	516	517

Table 9: *Template identifiers for T7 RDI/EMDI/MDI*

Note: The Template ID for the *packet header* will change in future releases and can be used to identify the software release. Up to 5 different *packet header* Template IDs are reserved for this purpose per interface (RDI, EMDI and MDI). It will change only in the event of changes in the interface e.g. a change in the XML FAST Templates (see 11 Detailed data feed description and layout).

Example: The TID=67 indicates the *packet header* for T7 MDI in the current release. If there are any interface changes in the next release, the TID for the *packet header* will change to another value.

5.2.4 Dictionaries

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

5.2.5 Stop bit encoding

Most FAST fields are stop bit encoded, each byte consists of seven *data bits* for data transfer and a *stop bit* to indicate the end of a field value. An exception from this rule are Byte Vectors as they are used in

the *packet header* of T7 EMDI/MDI/RDI.

5.2.6 FAST operators

Field operators are used to remove redundancies in the data values. Message templates are the meta-data for the message and are provided earlier. When the messages arrive, the receiving application has complete knowledge of the message layout via the template definition; it is able to determine the field values of the incoming message.
The following FAST operators are used in T7 EMDI/MDI/RDI:

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

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

5.3 Decoding the FAST-message

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

A typical example for a XML FAST template with a repeating group is shown in figure 19 of section 14.1, Example for a XML FAST template.

5.4 Transfer decoding

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

5.5 Composing the Actual FIX-Message

A typical FAST decoder would not deliver FIX messages after the decoding process. In order to compose FIX messages, applications need to apply additional rules.

The sequence of FIX-fields after composing the FIX-message on participants' side is not governed by the FIX-layout of the messages, i.e. the fields names of the FIX-message do not need to be in the same sequence. The FIX message, however, needs to fulfill the minimum requirement:

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

5.6 New features in FAST version 1.2

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

- **New Type Definition Syntax**: This allows the separation of the “type definitions” from the “type usage” within template definitions.
- **Enumeration**: This feature can be used when there is a fixed set of valid values for a single field.
- **Set (multi-value field)**: This feature can be used when there is a fixed set of valid values which could be sent together as a bit combination instead of using a repeating group. An example for a set would be the field *TradeCondition (277)* in the Depth incremental message. Sets are used to define the valid values for fields.
- **Timestamp Data Type**: The use of this feature allows native support of time stamp fields which becomes increasingly important for the T7 market data interface. A time stamp is an integer that represents a number of time units since an epoch.

5.7 Data types

The T7 implementation of FAST utilizes the following FAST data types:

- Decimal
- Length
- String
- uInt32/int32/uInt64/int64
- Byte vector
- Set
- Enum
- Timestamp

5.8 FAST version 1.1 compatible templates

Participants who choose not to upgrade their FAST 1.2 decoders can use FAST 1.1 compatible files offered by T7 trading architecture. The following needs to be considered:

- **Enumerations**: As described in the previous chapter enumerations have a list of codes. Participants receive an integer but not the description (meaning) of the integer. Since FAST 1.1 does not support enumerations this description of codes needs to be taken from the valid values provided with T7 Market and Reference Data Interfaces - XML FAST Templates.
- **Sets**: Similar to enumerations, however, participants receive a bitmap and multiple items from the list. The items need to be taken from the valid values provided with T7 Market and Reference Data Interfaces - XML FAST Templates.

The T7 Market and Reference Data Interfaces - XML FAST Templates files could be found at

www.eurex.com > Support > Initiatives & Releases > T7 Release 13.1 > System Documentation > Market & Reference Data Interfaces

or

www.xetra.com > Technology > T7 trading architecture > System documentation > Release 13.1 > Market and Reference Data Interfaces.

The FAST version 1.2 Extension Proposal describes how the encoded field (wire format) value looks.

Example for enumeration: *TradingSessionID* (336) can have one of the following values as defined in the FAST 1.2 XML files:

```
<define name="TradingSessionID">
  <enum>
    <element name="1" id="Day"/>
    <element name="3" id="Morning"/>
    <element name="5" id="Evening"/>
    <element name="6" id="AfterHours"/>
    <element name="7" id="Holiday"/>
    <copy/>
  </enum>
</define>
```

The wire format of the values 1, 3, 5, 6, 7 is 0, 1, 2, 3, 4, i.e. each value is represented by an index. Enumerations are not defined in the FAST 1.1 XML files. When the decoder receives a 4 he needs to know that it means “Holiday”.

Example for set: *TradeCondition* (277) can have one or more values as defined in the FAST 1.2 XML files:

```
<define name="TradeConditionSet">
  <set>
    <element name="U" id="ExchangeLast"/>
    <element name="R" id="OpeningPrice"/>
    <element name="AX" id="HighPrice"/>
    <element name="AY" id="LowPrice"/>
    <element name="AJ" id="OfficialClosingPrice"/>
    <element name="AW" id="LastAuctionPrice"/>
    <element name="k" id="OutOfSequenceETH"/>
    <element name="BD" id="PreviousClosingPrice"/>
    <element name="a" id="VolumeOnly"/>
    <element name="BC" id="TradingOnTermsOfIssue"/>
    <element name="SA" id="SpecialAuction"/>
    <element name="TC" id="TradeAtClose"/>
    <element name="XR" id="Retail"/>
  </set>
</define>
```

The wire format of the values U, R, AX, AY, AJ, AW, k, BD, a is 1, 2, 4, 8, 16, 32, 64, 128, 256, i.e. each value is represented by a different bit. The values can be added together to form combinations of the values. If U, AX are sent then 1 + 4 = 5 are the encoded field values.

Sets are not defined in the FAST 1.1 XML files. When the decoder receives a 5 he needs to know that it is a combination of 1 and 4 which is “ExchangeLast” and “HighPrice”.

6 Description of a typical trading day

This chapter describes a typical trading day, from the start until the end of trading; the following steps need to be taken to prepare for and to receive market data:

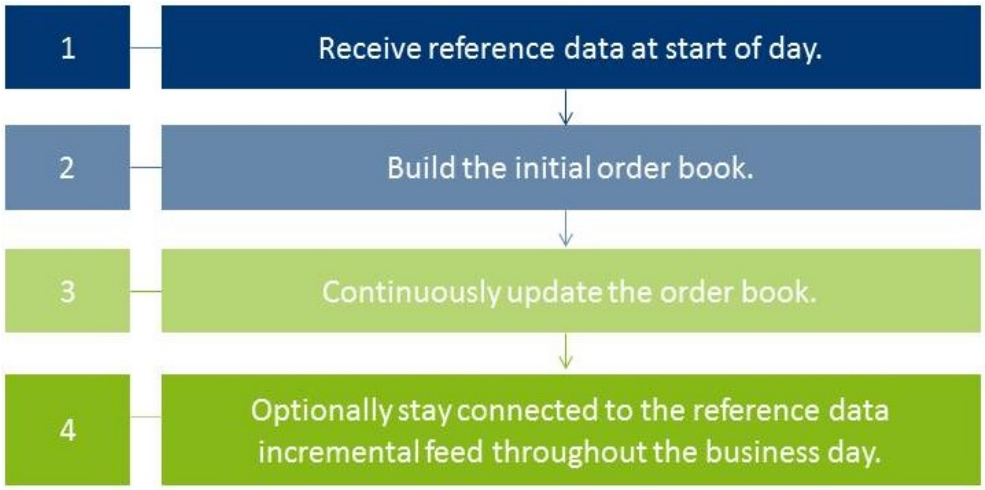

Figure 5: *Typical trading day*

6.1 Start of day operation

Before processing any market data, receiving applications need to retrieve technical and functional information via the T7 RDI. Alternatively, reference data can be received in file format (Reference Data from File). A detailed description of the reference data feeds and messages is provided in section 9.1, Reference data messages.

Members are advised to verify that the received reference data refers to the correct business day to ensure that the reference data processed on their end is actually the reference data for the business day in question and not reference data from e.g. the previous business day.

At start-up, reference data must be processed to create the initial order book baseline.

6.2 Receiving reference data via T7 RDI at start of day

At the start of a new business day, receiving applications need to join the static multicast address/port of the reference data interface in order to receive the following messages:

- **Product snapshot** to receive the functional and technical parameters.
- **Instrument snapshot**, **Variance futures status**, **Total return futures status** and **Trade at reference price status** messages to receive instrument details.
- **Instrument incremental** to receive add and modify of complex and flexible instruments.

Port information and multicast addresses for the reference data feeds as well as the address ranges for market data are published in the document “Network Access To Exchange Applications” and in section 12, Multicast addresses. Port information and multicast addresses for market data feeds are delivered as part of the reference data feeds.

Further detailed information about reference data is provided in section 9.2, General reference data rules. However, the basic steps in order to receive reference data are the following:

1. Listen to the reference data incremental feed and start buffering the messages. If an application starts listening to the reference data messages early enough, there are no messages available.
2. Listen to the reference data snapshot feed. Ignore all the messages until you reach the *Market Data Report* message denoting the beginning of a snapshot. Take note here of two values:
   - *MDReportCount*, containing the number of reference data snapshot messages in the initial snapshot cycle.
   - *LastMsgSeqNumProcessed*, containing the sequence number of the last message at the end of the snapshot cycle; This could be a snapshot or an incremental message.

   Process all messages of the snapshot until you encounter the *market data report* message denoting the end of the snapshot cycle.
3. At this point you need to complete the list with the messages received on the incremental feed since you started listening. But only after you have discarded all messages having[9]:

   MsgSeqNum <= LastMsgSeqNumProcessed - MDReportCount.
4. Store the reference data information for future use.
5. Join the market data incremental feed of EMDI or the market data feed of MDI in order to receive additional reference data changes.
6. Leave the reference data snapshot and incremental feeds.

Note: Applications starting early do not require steps 1 and 3 since no incremental message exists at this time.

New complex instruments predefined by the exchange are also sent in *Instrument incrementals* at partition startup before the start of the actual trading.

Note: Participants interested in complex or flexible instruments should use the *Complex instrument update* and *Flexible instrument update* messages. These are published on the market data incremental feed of the T7 EMDI as well as on the market data feed of the T7 MDI. They are published faster than the instrument incremental message of the reference data incremental feed.

6.3 Receiving reference data file (RDF) at start of day

Participants with low bandwidth connections may retrieve the start of day reference data in a file based format.

The initial reference data file generated at the start of a new business day contains the “reference data snapshots” available from the previous day without complex (and scaled simple) instruments. During the actual trading multiple incremental files are created as complex and flexible instruments are added.

Complex (and scaled simple) instruments from the previous day and new complex instruments predefined by the exchange are sent in incremental files after the T7 product state change to Start of Day (see Product State Changes 4.2.1) or after a restart of T7 RDI (RDF).

In case a receiving application starts late, each of the intraday Reference Data Files in addition to the Start-Of-Day Reference Data File must be applied. Start-Of-Day and Intraday Reference Data Files are available via the *Common Report Engine*.

Note: In case a late starting application uses the Start-Of-Day Reference Data File without the intraday files, the intraday created complex instruments remain unknown and hence order book data may be received for unknown instruments.

[9] The snapshot and incremental feeds have a different sequence number range.

6.4 Build the initial order book

Participants first have to build the initial order book. The order book has to be maintained per instrument.

Note: Sequence numbers contained in the market data messages are incremented per product.

6.4.1 Build the initial order book with the T7 EMDI

For each instrument within the desired products do the following:

Figure 6: *T7 EMDI initial order book*

6.4.2 Build the initial order book with the T7 MDI

The following sequence is recommended for the **T7 MDI:**

Figure 7: *T7 MDI initial order book*

The field *LastMsgSeqNumProcessed (369)* in the T7 MDI snapshots can be ignored because snapshots and incrementals are sent in-band and don't need to be synchronized with each other.

Note: Since T7 Release 8.0 MDI might not send all snapshots of all instruments of a product contiguously. T7 MDI incremental messages might contain incremental entries (MDIncGrp) for all instruments of a product. A joining application which is in the middle of building the initial order books *must* discard entries for instruments for which they have not received a snapshot yet.

6.5 Update the order book

Every update in the form of a *Depth incremental* or *Depth snapshot* message contains the price level and the actual price to which the instruction needs to be applied. The receiver application can update information at a particular level with the new information.

Once participants have built the current order book it needs to be continuously updated:

6.5.1 Update the order book with the T7 EMDI

As long as the *MsgSegNum* values for the *Depth incremental* message are contiguous per product do the following [10]:

- Keep applying all *Depth incremental* messages to the current order book.

Note: *Depth snapshot* messages are sent on a different channel to the *Depth incremental* messages. Changes to the order book are also sent using the *Depth snapshot* messages but the information is also provided with the incremental messages. Snapshot messages don't need to be processed unless the order book needs to be recreated.

[10] The reason is that the unreliable nature of UDP multicast can cause packets to arrive delayed, in incorrect sequence or may be missing.

6.5.2 Update the order book with the T7 MDI

As long as the *MsgSegNum* values for the *Depth incremental* message are contiguous per product do the following[10]:

- Keep applying all *Depth incremental* as well as *Depth snapshot* messages (with *RefreshIndicator (1187) = Y*) to the current order book.

Each incremental message can carry different update instructions with the “update action” (New, Change, Delete, Delete From, Delete Thru, Overlay).

Note: The *Depth snapshot* messages for the T7 MDI are sent on the same channel as the *Depth incremental* messages. If the *RefreshIndicator (1187)* is set, changes to the order book are processed into the *Depth snapshot* messages and not provided as separate *Depth incremental* messages.

7 Recovery

Due to the unreliable nature of UDP multicast it is possible that some packets may either be delayed, arrive in the incorrect order or even be missing. Furthermore the UDP packets may be duplicated at the network level. Receiving applications need to be capable of handling these issues. This chapter describes the scenarios which might occur and provides a guideline on how a receiving application needs to react to those scenarios.

Recovery actions are possible on a packet level by using the respective other service (A or B). In case a packet is lost on both services (A and B) clients can create a new current order book by using snapshot information.

7.1 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 is achieved by a simple memory comparison on the first 10 bytes for T7 EMDI or 8 Bytes for T7 MDI of a datagram containing *SenderCompId* and *PacketSeqNum* as shown in figure 17, Structure of the packet header for T7 EMDI and figure 18, Structure of the packet header for T7 MDI and T7 RDI. Another important function of the *packet header* is to identify **gaps** by means of the *PacketSeqNum* which can be retrieved just by decoding the *packet header*.

Note: Packets with the same *SenderCompID (49)* have contiguous sequence numbers per multicast address / port combination.

This means that field *PacketSeqNum* can be used not only to detect duplicates but also to detect missing packets. *PacketSeqNum* is a Byte vector and therefore not stop bit encoded as per the FAST specification.

The *packet header* itself does not contain any product information. In order to find out which product is missing, the product level sequence number must be used in addition to the packet level sequence number; the packet needs to be decoded further down to the message level. This leaves participants with **two recovery options** when a gap in the *PacketSeqNum's* of the *packet header* is detected.

Example:
A single multicast address carries products FDAX and FGBL, but the participant is only interested in FGBL.

I. Pessimistic approach: The receiving application assumes that FGBL is part of the missing packet: It immediately starts recovery actions [11] just by decoding the *packet header*.

- **Advantage:** Recovery is triggered immediately when observing a missing *PacketSeqNum* without decoding the entire message.
- **Disadvantage:** The recovery might not be necessary, if FGBL is not part of the message which is inside the lost packet.

II. Optimistic approach: The receiving application assumes that FGBL is not part of the missing packet: It waits for the next message on the same service and decodes the packet up to the message level to find out if a packet for FGBL has been lost before triggering recovery actions.

- **Advantage:** This approach allows the participant to recover only products of interest.
- **Disadvantage:** The receiving application needs to wait for the next message. However, the next packet may not contain a message for the product in question.

[11] by means of the other service (live-live concept) or by listening to the *depth snapshot*

7.2 How to recover data via the respective other service (A or B)

Feeds are replicated onto two services, “Service A” or “Service B”, and carried on different multicast addresses. This feature provides the possibility to recover missed packets, and participants are advised to join both services.

In each of the following tables, the “Time” column is entirely arbitrary and is intended to show only the sequence of events and in some cases the relative delay between dependent events.

The following table explains the design concept for Service A and B. The table contains the field *MsgSeqNum* from the message itself. However, it could also contain the field *PacketSeqNum* from the *Packet Header*.

	Service A:			Service B:	
Time	MsgSeqNum	Message	Time	MsgSeqNum	Message
10:30:00	206	New 151@4	10:30:01	206	New 151@4
10:30:05	207	Delete 151@5	10:30:07	207	Delete 151@5
	lost		10:30:12	208	New 151@5
10:30:10	209	New 152@4	10:30:13	209	New 152@4

Table 10: *Recovery via Service B (live-live concept)*

As the above example shows, the same information is delivered on Service A and B. While *MsgSeqNum* = 208 is missing on Service A, it is provided on Service B.

Ideally a receiving application processes packets from both Service A and B simultaneously and would take into account the message that arrives first and discardes the second (identical) message.

In the unlikely event that the message has neither been received via Service A nor Service B, the receiver is required to initiate a loss of data scenario:

- The order book needs to be recreated by using the **depth snapshot** messages in conjunction with the **depth incremental** messages. This procedure is similar to the Start Up procedure. Please see section 6.4, Build the initial order book.
- The maximum expected recovery interval for a particular feed can be obtained in the *Product snapshot* message of the T7 RDI snapshot feed (field: *MDRecoveryTimeInterval (2565)*).

7.3 Delayed packets

The following example indicates a simple case:

Time	MsgSeqNum	Message
10:30:00	132	New 151@4
10:30:04	133	Delete 151@5
10:30:39	134	New 152@4

Table 11: *Packets arriving in correct sequence*

In this example, messages arrive in the correct order. The message was not delayed between T7 and the receiving application. There is no special requirement on the application; the message can be processed in the same order as they arrive.

Multicast does not guarantee that the order in which packets are received is the same as the order in which they are sent. For instance, T7 Market Data Interface sends incremental messages in ascending *MsgSeqNum* order, but they might arrive in an incorrect order at the receiving application.

Consider the following example:

Time	MsgSeqNum	Message
10:30:00	206	New 151@4
10:30:04	208	Delete 151@5
10:30:10	207	New 152@4

Table 12: *Delayed Packet 207*

In this example, message 207 is delayed within the network, allowing message 208 to arrive first.

A correct communications layer responds as follows:

1. Release message 206 to the application immediately on arrival.
2. On arrival of 208, recognises that 207 is missing.
3. Start an appropriate timed operation to trigger the recovery actions if the out-of-sequence message 207 fails to arrive in a reasonable time.
4. Assuming that 207 arrives within that reasonable time, release 207 and then 208 to the application in that order and cancel the timed recovery action.

7.4 Missing packets

All lost packets start life as “delayed” packets, as illustrated in the preceding case. The communications layer of the receiving application is responsible for deciding when to declare a network packet as lost. In the following example it is assumed that *MsgSeqNum* = 207 from the example above does not arrive within the allowed time. Therefore it is considered as lost:

Time	MsgSeqNum	Message
10:30:00	206	New 151@4
	lost	
10:30:04	208	Delete 151@5
10:30:10	209	New 152@4

Table 13: *Missing seqNum 207*

The correct behaviour in this instance is:

1. Release message 206 immediately on arrival.
2. Hold on to 208 because it is out-of-sequence, and initiate timer-based recovery actions.
3. Hold on to 209 for the same reason. Timer-based recovery actions are already pending for this product, so do not reset the timer.

(a) Even though message 209 is a "New" operation, it may be unsafe to apply 208 and 209 because we do not know what 207 contains.

4. If the missing message (207) fails to arrive within the allowed time:

   (a) Initiate recovery from the respective other service (A or B) for message (207). If this works then release (207) and then all messages with higher *MsgSeqNum's*.

   (b) In case the recovery from the respective other service (A or B) fails: initiate recovery via snapshots.

7.4.1 Recovery (T7 EMDI)

Depth snapshot and *depth incremental* messages are distributed via separate channels for the EMDI. For instance, *depth incremental* messages could be sent on multicast address $A_2{}^I$, port x and the snapshot message on multicast address $A_2{}^S$ with port y (see Figure 2, Overview of the three interfaces).

Incrementals are sent whenever there is a change of the order book (event-driven); snapshots are sent periodically in intervals regardless of whether the order book has changed since the last snapshot (time-driven).

Each message sequence number (field: *MsgSeqNum*) on the market data incremental feed is unique and contiguous by product across messages. Therefore the sequence number can be used to detect losses. If any gap of the arriving sequence numbers is detected and this gap cannot be filled by using the respective other service (A or B) the receiving application should initiate a snapshot recovery.

The following example shows missing *depth incremental* messages (*MsgSeqNum's* 208-209) and depth snapshots (with *LastMsgSeqNumProcessed*) which relate to the missing message. *MsgSeqNum's* for the depth snapshot do not exist, which is indicated with "N/A" in the table.

MsgSeqNum	Product	LastMsgSeq-NumProcessed	Message Type	Channel
205	A		quote request	$A_1{}^I$
206	A		depth incremental	$A_1{}^I$
207	A		depth incremental	$A_1{}^I$
lost	A		depth incremental	$A_1{}^I$
lost	A		depth incremental	$A_1{}^I$
210	A		depth incremental	$A_1{}^I$
1000	B		depth incremental	$A_2{}^I$
N/A	A	209	depth snapshot	$A_1{}^S$
211	A		depth incremental	$A_1{}^I$
N/A	B	1000	depth snapshot	$A_2{}^S$
1001	B		depth incremental	$A_2{}^I$

Table 14: *Snapshots and incrementals within the T7 EMDI*

The appropriate recovery action for missing *depth incrementals* is the same as the logic described in section 6.4.1, Build the initial order book with the T7 EMDI.

There are some additional points to be aware of when performing recovery:

- During recovery, applications should be prepared to receive *depth incremental* messages for instruments they didn't know existed. This can occur if a strategy creation event (via a *complex instrument*

update on the market data feed) is missed due to packet loss. In this case, applications must consult the reference data snapshot feed to obtain the strategy description.

- *Depth snapshot* messages are not sequenced, but they are still theoretically subject to out-of-order packet delivery. Applications must consider this in determining that their snapshot cycle is complete. The packet sequence number in the *packet header* can be used to detect out-of-order delivery.
- The *LastMsgSeqNumProcessed (369)* is not necessarily the same for all instruments belonging to a product on the market data snapshot feed.

Note: The market data snapshot feed does not contain any “start” or “end” messages to delineate the cycle.

There are two ways to determine when to leave the snapshot feed during recovery:

Method 1: Process specific products

For each *SenderCompID (49)* contributing to the market data snapshot feed, *depth snapshot* messages are grouped by product as illustrated below:

P_1I_1 | P_1I_2 | P_1I_3 | P_1I_n | P_2I_1 | P_2I_2 | P_2I_3 | P_2I_n | P_3I_1 | P_3I_2 | P_3I_3 | P_3I_q | [...]

with:
P_n: Product n
I_q: Simple instrument q for product n

Depth snapshots for instruments in the same product will often all appear in the same packet, but this should not be relied upon as it is not true when the amount of data is simply too great to fit into a single packet, and under certain other technical conditions on the exchange.

A change of product *MarketSegmentID (1300)* for a given *SenderCompID (49)* indicates the end of the *depth snapshot* messages for the respective product. This allows applications to easily determine when they’ve received a snapshot for every instrument in the products they’re interested in and leave the snapshot feed.

Method 2: Process an entire depth snapshot cycle

It's also easy for an application to listen to an entire snapshot cycle.

Applications can determine when they've seen an entire snapshot cycle simply by remembering the *SecurityID (48)* of the first *depth snapshot* message they saw from each *SenderCompID (49)*.

When they see the same *SecurityID (48)* again for each *SenderCompID (49)*, they know that a complete depth cycle has been seen and can leave the snapshot feed.

Note: Receiving applications also need to consider *depth snapshot* messages for newly created complex instruments.

Note: If the *SenderCompID (49)* for the affected partition changes during snapshot processing, applications should listen to a new snapshot cycle for that partition again (see 9.10.3).

7.4.2 Recovery (T7 MDI)

Snapshot and incremental messages are sent on the same channel and carry a contiguous sequence number (field: *MsgSeqNum*) per product. The snapshot always carries the latest information and might carry new information, not already sent with an incremental message. The following table shows an example for the distribution of incremental and snapshot messages for two products:

MsgSeqNum	Product	Message Type	Channel
5	A	quote request	$A_1{}^{S,I}$
6	A	depth incremental	$A_1{}^{S,I}$
lost	A	depth incremental	$A_1{}^{S,I}$
25	B	depth incremental	$A_2{}^{S,I}$
8	A	depth incremental	$A_1{}^{S,I}$
9	A	depth snapshot	$A_1{}^{S,I}$
10	A	depth snapshot	$A_1{}^{S,I}$
11	A	depth incremental	$A_1{}^{S,I}$
26	B	depth snapshot	$A_2{}^{S,I}$
27	B	depth incremental	$A_2{}^{S,I}$

Table 15: *Snapshots and incrementals within the T7 MDI*

If the *depth incremental* message for product A with *MsgSeqNum* = 7 is lost, a consistent order book can be rebuilt from the next snapshot message for product A, in this case arriving with *MsgSeqNum*=9.

All depth incremental messages for product A with a lower sequence number than the next market data snapshot message for product A must be discarded, e.g. *MsgSeqNum* = 8 (incremental) must be discarded as its effect is included in MsgSeqNum = 9 (snapshot).

Since multicast doesn't guarantee the correct sequence of the incoming message, it is recommended to buffer all incoming incrementals while waiting for the next snapshot message. The buffered incrementals for product A with *MsgSeqNum* ≥ 11 can be applied to the latest snapshot with *MsgSeqNum* = 10.

Note: *LastMsgSeqNumProcessed* is not necessary for recovery purposes in the T7 MDI.

8 Various time stamps in T7 and how to use them

The various 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 T7 EMDI, T7 MDI and T7 EOBI feeds. These can be used to analyze one way transport times. To reiterate, 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 the following locations:,

Gateway:

- On entry to the Gateway.

MatchingEngine:

- 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 T7 ETI Manual).

Market Data (T7 EMDI, T7 MDI and T7 EOBI):

- *SendingTime* for order book delta and snapshot messages,
- additionally, timestamps from Matching Engine such as *Matching Engine In* timestamp, *Priority-Timestamp* or *TransactTime* and *Gateway In* timestamp, etc. are provided on market data messages.

The following picture provides an overview of T7 timestamps:

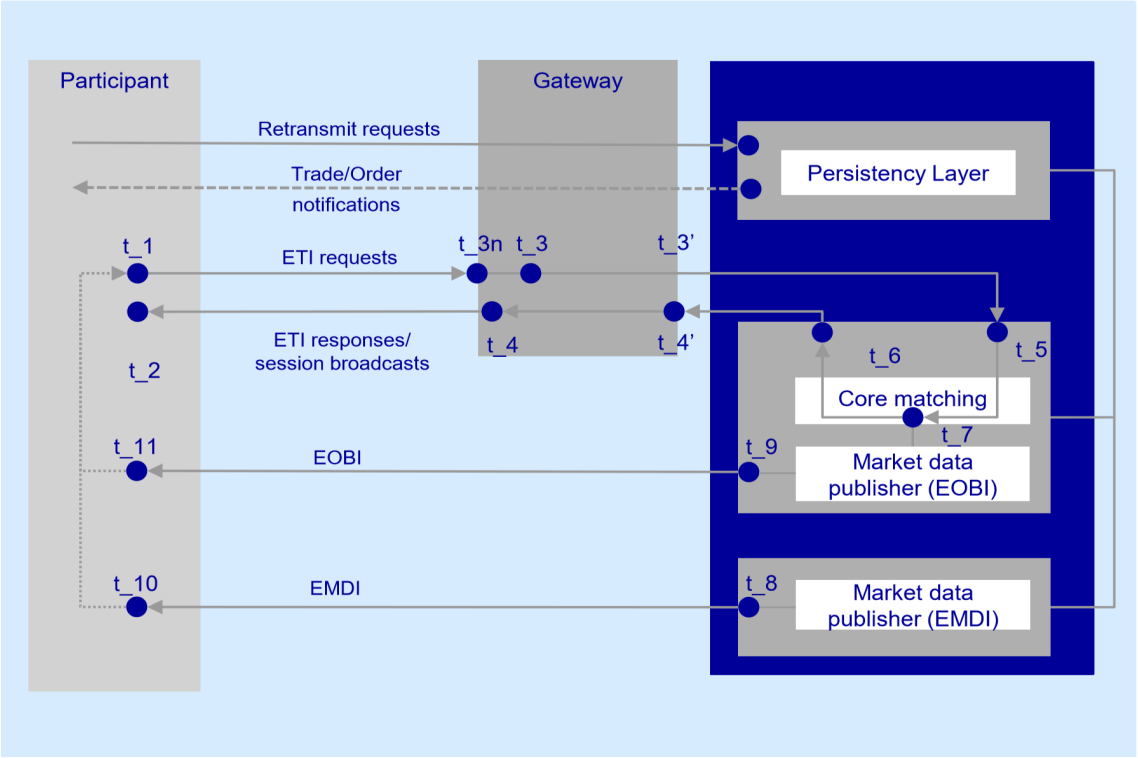

Figure 8: *Timestamp Overview*

The following table lists the mapping of T7 timestamps:

Timestamp	Semantic	FIX fields	Description
t_3n	Gateway request in	*If available:* RequestTime (5979)	Time taken by the ETI gateway when the first bit of a request arrives on the PS gateway NIC.
t_3	Gateway request in	*If t_3n is not available:* RequestTime (5979)	Time taken by the ETI gateway application when a request is read from the socket on the Participant's side of the gateway.
t_5	Matching engine in	AggressorTime (2445)	Time taken by the matching engine when a request is read.
t_7	Priority timestamp, Creation timestamp, Transaction timestamp, etc.	TrdRegTSTimePriority (21008), ExecID (17), TransactTime (60), MDEntryTime (273), etc.	Time taken when a transaction is functionally processed. It 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_8	T7 EMDI out	SendingTime (byte vector)	Provides the sending time when T7 EMDI has put the datagram on the wire.
t_9	T7 EOBI out	TransactTime (60)	Provides the sending time when T7 EOBI has put the datagram on the wire.

Table 16: *Timestamp mapping*

9 Important topics with use cases and examples

The following section “Use Cases” describes situations which require special attention. Various examples are provided.

9.1 Reference data messages

Reference data provides technical and functional information about all products and instruments available in T7. Reference data messages are sent within different feeds:

- **Snapshot feed of T7 RDI** provides a snapshot of all products and instruments (simple, complex and flexible) and is sent out on a regular basis throughout the day. Additions of complex and flexible instruments are incorporated into the next snapshot cycle.
- **Incremental feed of T7 RDI** is event triggered and provides real-time information about complex instruments[12] scaled simple and flexible instruments that are added intraday and about variance futures, total return futures and trade at reference price status updates. Any change is incorporated within the next snapshot cycle.
- **Market data incremental feed of EMDI** is event triggered and provides real-time information about complex and flexible instruments that are added or inactivated intraday on the same channel as market data.
- **Market data feed of MDI** is event triggered and provides real-time information about complex and flexible instruments that are added or inactivated intraday.

The following messages are sent via different feeds:

a) **Snapshot feed of T7 RDI**:

- ***Product snapshot*** for products available at start of day.
- ***Instrument snapshot*** for simple and flexible instruments available at start of day.
- ***Variance futures status*** for variance futures instruments.
- ***Total return futures status*** for total return futures instruments.
- ***Trade at reference price status*** for trade at reference price instruments.
- ***Instrument incremental*** for complex and flexible instruments.
- ***Market data report*** indicates the start of reference data (*MDReportEvent=1*).
- ***Market data report*** indicates the end of reference data (*MDReportEvent=2*).

b) **Incremental feed of T7 RDI**:

- ***Instrument incremental*** for complex and flexible instruments.
- ***Variance Futures Status*** when the conversion parameters have been approved.
- ***Total Return Futures Status*** when there is a change in any of the conversion parameters.
- ***Trade At Reference Price Status*** when there is a change in any of the conversion parameters.

c) **Market Data incremental feed of EMDI**:

- ***Complex instrument update*** for complex instruments added intraday.

[12] No product information is delivered

- ***Flexible instrument update*** for flexible instruments added intraday.
- ***Instrument state change*** or ***mass instrument state change*** for complex instruments inactivated or re-activated intraday.

d) **Market data feed of MDI**:

- ***Complex instrument update*** for complex instruments inactivated or re-activated intraday.
- ***Flexible instrument update*** for flexible instruments added intraday.

9.2 General reference data rules

9.2.1 General structure of the snapshot cycle

A snapshot cycle consists of (see figure 9):

- A ***Market Data Report*** message (*MDReportEvent* = 1 = **StartOfReferenceData**).
- A sequence of a *Product snapshot* followed by the associated *Instrument snapshot* messages, repeating for all products and instruments.
- A dynamically growing sequence of *Instrument incremental* messages.
- Finally ***Market Data Report*** message (*MDReportEvent* = 2 = **EndOfReferenceData**).

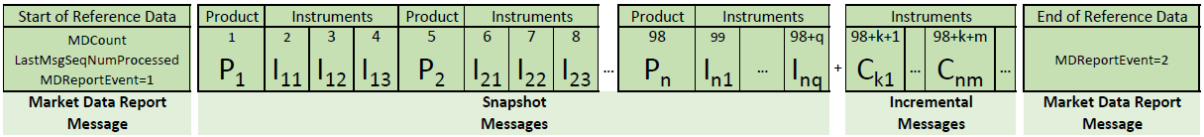

Figure 9: *Entire snapshot cycle on the T7 RDI snapshot feed*

with:
P_n: Product n
I_{nq}: Simple instrument q for product n
C_{nm}: Flexible or complex instrument m for product n

Product and *Instrument snapshot* messages are sent for the initial set of products and instruments. While the snapshots do not change intraday, the number of incremental messages increases if, e.g., complex and flexible instruments are added. Figure 10 illustrates how more *Instrument incrementals* are added over the course of n cycles:

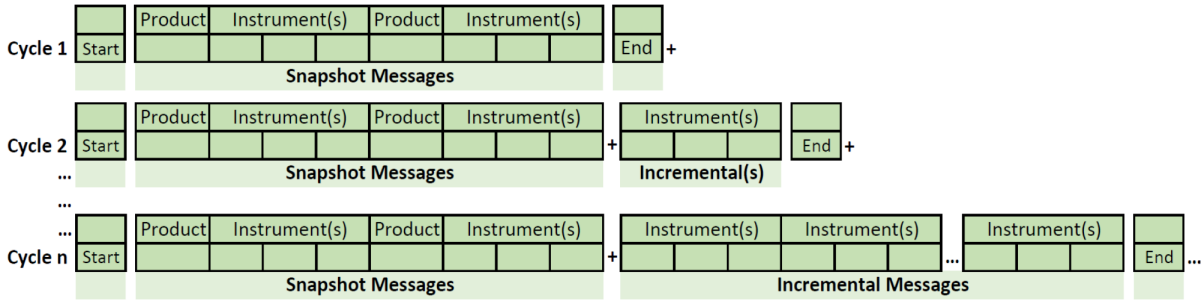

Figure 10: *Reference data with constant snapshots and extending incrementals*

Note: Overnight changes to the products and instruments are reflected in the *reference data snapshot messages* after the technical start on the next business day.

9.2.2 Counters as part of the market data report message

The message sequence numbers of the *Market Data Report* messages preceding each snapshot cycle represent counters for the number of snapshots, incrementals and overall number of messages within the current cycle. The *Market Data Report* message of type **StartOfReferenceData** contains the following sequence number fields:

- ***MDReportCount (2536)***: Number of reference data messages in the snapshot cycle, which is available at the start of day and remains constant throughout the operational hours of reference data service for the current business day. The value represents the number of product level messages + the number of instrument level messages (simple instruments, variance futures states, total return futures states and trade at reference price states) at the start of day.

  If a failure of T7 RDI occurs the number of messages in the reference data snapshot and herewith *MDReportCount* (2536) may change.

- ***LastMsgSeqNumProcessed (369)***: This is the *MsgSeqNum* value of the last reference data message (snapshot or incremental) in the snapshot cycle (products and instruments share a single sequence number).

  Note: The number of incremental updates in a snapshot cycle can be calculated as:
  Number of incremental updates = *LastSeqNumProcessed* - *MDReportCount*.

- ***TotNoMarketSegmentReports (2537)***: Contains the number of product level messages sent in the snapshot cycle. This value remains constant intraday as products are not created or deleted intraday.

- ***TotNoInstrumentReports (2538)***: Contains the number of instrument level messages sent in the snapshot cycle. This value changes as more flexible and complex instruments are created intraday or as variance futures status, total return futures status and trade at reference price status messages are disseminated.

TotNoMarketSegmentReports (2537) and *TotNoInstrumentReports (2538)* can be used as a sanity check and to pre-allocate the product and instrument containers.

The *Market Data Report* message of type **EndOfReferenceData** marks the end of reference data messages and does not contain any counters.

The following examples highlight a few scenarios which require special attention. The focus lies on the reference data snapshot feed which provides constant snapshot messages and a variable part with incrementals for flexible[13] and complex[14] instruments.

[13] Participants interested in flexible instruments can also use the *Flexible instrument update message* via the faster depth incremental feed of EMDI or the market data feed of MDI.

[14] Participants interested in complex instruments can also use the *Complex instrument update message* via the faster depth incremental feed of EMDI or the market data feed of MDI.

9.2.3 Use case 1: Reference data at the start of the reference data service

At the **start** of the reference data service the *reference data snapshot* is sent. Figure 11 shows how snapshots for simple instruments, which are already in the system, are sent at start of day:

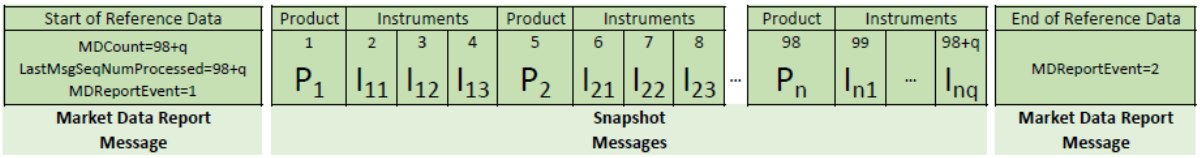

Figure 11: *Reference data snapshot messages on the reference data snapshot feed at the start of the reference data service*

The initial snapshot cycles do not contain any flexible or complex instruments in the snapshot messages. A reference data *Instrument incremental* message does not exist at this time.

9.2.4 Use case 2: Reference data after intraday addition of complex instruments

The next example shows an **intraday** addition of three complex instruments C_{11}, C_{41} and C_{31}. See figure 12). The reference data *Instrument incremental* messages for complex instruments C_{11}, C_{41} and C_{31} are appended to the reference data *Instrument snapshot* messages:

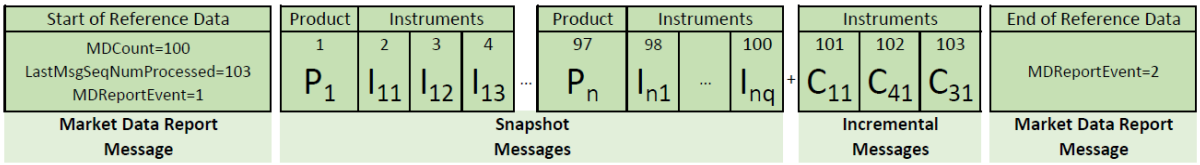

Figure 12: *Reference data snapshot after intraday addition of complex instruments C_{11}, C_{41} and C_{31}*

LastMsgSeqNumProcessed (369) in the *Market Data Report* message of type **StartOfReferenceData** increases to 103. The number of incremental messages can be calculated as *LastMsgSeqNumProcessed* - *MDReportCount* = 103 - 100 = 3.

New complex instruments predefined by the exchange are sent as *Instrument incremental* messages and not as *Instrument snapshot* messages on the day of creation; the messages are sent before trading starts.

9.2.5 Use case 3: Reference data on the next business day

The complex instruments which still exist on the next business day and which have been sent as reference data *Instrument incremental* messages on the previous business day, are sent again as reference data *Instrument incremental* messages on the next business day on T7 product state change to Start of Day (see Product State Changes 4.2.1) or after a restart of T7 RDI. These incrementals have *TransactTime* equal to creation time as of the last business day.

9.2.6 Use case 4: Failover or restart of T7 RDI

In the event that a T7 RDI fails, another instance takes over. Receiving applications can detect this by a change of the *SenderCompID* for a marketID and the receipt of a *Market Data Report* message. Applications should respond to this situation as described in section 6.2, Receiving reference data via T7 RDI at start of day. The same recovery actions apply in case of a complete restart of T7 RDI.

9.2.7 Use case 5: Chronological order of messages for complex instrument creation

The creation of complex / flexible instruments results in the following sequence of messages:

1. On the **market data incremental feed** of the T7 EMDI and **market data feed** of the T7 MDI: A *Complex instrument update* (alternatively *Flexible instrument update*) message is sent to inform the participant as fast as possible. In case a new complex / flexible instrument has been created the corresponding message is sent prior to the publication of any order book data for the new complex / flexible instrument.
2. On the **reference data incremental feed** of the T7 RDI: An *Instrument incremental* message is also sent with additional fields populated. There is no *Product incremental* message.
3. On the **reference data snapshot feed** of the T7 RDI: An *Instrument incremental* message is appended to the end of the current snapshot cycle without removing or changing any of the existing snapshot or incremental messages in the cycle[15]. Therefore the cycle is only extended intraday and never reduced.

9.2.8 Use case 6: Chronological order of messages for complex instrument deletion

For the deletion of a complex instrument, e.g. initiated by Market Supervision, an *Instrument state change* or *Mass instrument state change* message is sent with *SecurityStatus (965)* set to 2 = Inactive for the specific SecurityID on the **market data incremental feed** of the T7 EMDI and **market data feed** of the T7 MDI. No message is sent by the T7 RDI.

In case the complex instrument is re-created on the same day, an *Instrument state change* or *Mass instrument state change* message is sent with *SecurityStatus (965)* set to 1 = Active for the specific SecurityID on the **market data incremental feed** of the T7 EMDI and **market data feed** of the T7 MDI. No message is sent by the T7 RDI.

9.2.9 Use case 7: Variance Futures Status messages

At the start of day, a variance futures instrument appears on the **reference data snapshot feed** of the T7 RDI as a regular *Instrument snapshot* message immediately followed by a *Variance Futures Status* message. Therefore, for a variance futures instrument, field *TotNoInstrumentReports (2538)* in the *Market Data Report* is incremented by 2.

During the day, a *Variance Futures Status* message is sent on the **reference data incremental feed** of the T7 RDI when the conversion parameters have been approved. The same message is then appended to the end of the current snapshot cycle of the **reference data snapshot feed** of the T7 RDI.

9.2.10 Use case 8: Product pool

A Product Pool is a facility to enable functionalities which depend upon interaction of a set of products. Multiple products can be related by a product pool.

The product pool facility is introduced in the form of an IPS product pool, i.e. a type of product pool, the purpose of which is to support IPS functionality. An IPS product pool plays the role of a product for IPS instruments. I.e. IPS instruments do not belong to a product, but instead IPS instruments belong to an IPS product pool.

The *Product snapshot* message is used to distribute information about product pool parameter settings in the same way as information about product parameter settings. The MarketSegmentID field and the MarketSegment field identify the product pool in the same way as they identify a product. The other fields are filled for product pools, if they are available for product pools. For example, there is no underlying for

[15] A complete snapshot cycle is a combination of start, refdata snapshots, refdata incrementals and end message.

a product pool, so the fields related to the underlying of a product are not filled, but fields transporting price step tables are filled as they are set up for inter-product spreads on product pool level.

The ProductSnapshot message specifically does not contain information about which products are associated to the pool.

9.2.11 Use case 9: Flexible instruments

The intraday creation of flexible instruments results in the following sequence of messages:

1. On the **market data incremental feed** of the T7 EMDI and **market data feed** of the T7 MDI: a *Flexible instrument update* message is sent to inform the participant as fast as possible. In case a new flexible instrument has been created the corresponding message is sent prior to the publication of any order book data for the new flexible instrument.
2. On the **reference data incremental feed** of the T7 RDI: an *Instrument incremental* message is also sent with the following fields populated: *SecurityID (48)*, *CFICode (461)*, *SecurityDesc (107)*, *MinPriceIncrement (969) MinPriceIncrementAmount (1146)*, *ContractMultiplier (231)*, *ProductComplex (1227)* , *MaturityDate (541)*, *MaturityMonthYear (200)*, *SecurityExchange (207)*, *SecurityType (167)*, *PutOrCall (201)*, *StrikePrice (202)*, *StrikePricePrecision (2577)*, *OptAttribute (206)*, *ExerciseStyle (1194)*, *InstrumentPricePrecision (2576)*, *SettlMethod (1193)*, *SettlSubMethod (2579)*, *SecurityStatus (965)* and *MarketSegmentID (1300)*.
3. On the **reference data snapshot feed** of the T7 RDI: an *Instrument incremental* message is appended to the end of the current snapshot cycle without removing or changing any of the existing snapshot or incremental messages in the cycle[16]. Therefore the cycle is only extended intraday and never reduced.

9.2.12 Use case 10: Total Return Futures Status messages

At the start of day, a total return futures instrument is published on the **reference data snapshot feed** of the T7 RDI as a regular *Instrument snapshot* message immediately followed by a *total return futures status* message. Therefore, for a total return futures instrument, field *TotNoInstrumentReports (2538)* in the *market data report* is incremented by 2.

During the day, a *total return futures status* message is sent on the **reference data incremental feed** of the T7 RDI each time there is an update on any field of the message. The same message is then appended at the end of the current snapshot cycle of the **reference data snapshot feed** of the T7 RDI.

9.2.13 Use case 11: Trade At Reference Price Status messages

At the start of day, a trade at reference price instrument is published on the **reference data snapshot feed** of the T7 RDI as a regular *Instrument snapshot* message immediately followed by a *trade at reference price status* message. Therefore, for a trade at reference price instrument, field *TotNoInstrumentReports (2538)* in the *market data report* is incremented by 2.

During the day, a *trade at reference price status* message is sent on the **reference data incremental feed** of the T7 RDI each time there is an update on any field of the message. The same message is then appended at the end of the current snapshot cycle of the **reference data snapshot feed** of the T7 RDI.

[16] A complete snapshot cycle is a combination of start, refdata snapshots, refdata incrementals and end message.

9.3 General order book rules and mechanics

The T7 Market Data Interfaces, T7 EMDI and MDI, provide order book updates from level 1 to the maximum level. The maximum level is provided for each product in the *Product snapshot* messages in the reference data, field *MarketDepth (264)*. The order book can be constructed by the depth incremental messages or by the depth snapshot message.
All on-exchange trades and order book updates are reported via the same *depth incremental messages*. However, trades are always sent out prior to order book updates. The following design principles apply to order book updates:

- Orders are aggregated per price level and are not distributed individually.
- Changes to the book that result from one atomic action in the matching engine are disseminated in one *depth incremental* message for T7 EMDI.
- Each T7 EMDI packet relates only to a single product. In other words, although each T7 EMDI packet may contain multiple messages, those messages will always relate to the same product. This does not apply to T7 MDI where a single packet may relate to multiple products.
- Price levels are provided explicitly (field: *MDPriceLevel* (1023)) and do not need to be derived through the price itself.
- During the product states “Start-Of-Day“ and “Pre-Trading“, or when no price levels exist, an empty book (MDEntryType=J) is disseminated for the depth snapshot message (not for incremental). In “Pre-Trading“, statistical information is sent in addition to an empty book.
- During the product states “Post-Trading“ and “End-Of-Day“, ToB prices (MDEntryType 0=Bid and/or 1=Offer) are sent for simple and complex instruments using *depth incremental* messages. Furthermore, *depth snapshot* messages continue to disseminate the same during product states “Post-Trading“ and “End-Of-Day“.
- Valid for derivatives: An implied price is the only element of the group without a price level (for *MDEntryType (269)* 0 = Bid or 1 = Offer). For price levels from 1 to max. price levels, outright prices are distributed. An implied price can either be fully implied or partially implied (for more information please refer to section 9.3.1, Determination of the price sources).
- Valid for cash instruments: Depending on instrument configuration a surplus may be displayed during auctions. *MDEntryType (269)* is 0 = Bid or 1 = Offer, *MDPriceLevel (1023)* is not set and *QuoteCondition (276)* is set to Z = Order imbalance. A new surplus with *MDUpdateAction (279)* 0 = New overwrites the old surplus and side information (either Bid or Offer). For products for which the imbalance side is not disclosed, *MDEntryType (269)* A = Imbalance is used instead and *QuoteCondition (276)* will be empty.
- If two (or more) synthetic prices (with the same price) are created for the Best Market via a different path, then the quantity from the path with maximum quantity is reported for the particular price. An example is provided in section 14.2.4 table 57.
- Top Of Book information resulting from synthetic IPS matching opportunities is disseminated using the Top Of Book Implied message on the T7 EMDI incremental feed. There is no mechanism to guarantee that the Top Of Book messages are in the same datagram as the *depth incremental* messages. The same information is also sent using depth snapshot messages with the same MDEntry group but with MDBookType=Top Of Book(1) and MDSubBookType=Implied without restrictions(1) / Implied with restrictions(2). For details of the Top Of Book Implied message, see section 11.3.3, Top Of Book Implied message.
- There can be multiple updates in one message. The bid side is updated first followed by the ask side.
- If update instructions “new” or “delete” is sent for an implied price, the order book levels 1-n don’t need to be shifted down or up.

- Order book update instructions are sent for each order book side without a specific order of update actions but ordered by price level instead.
  - from best outright price (price level 1)
  - down to the worst price (max. price level configured per product).
  - if the resulting book depth, after each applied individual orderbook update instruction, is larger than the specified maximum product depth only the specified maximum product depth must be saved.
- For auctions, the best bid/ask prices are disseminated at price level 1 without a quantity. Receiving applications need to delete a pre existing quantity when an absent value is received during a transition into an auction.
- During an auction, there can either be a crossed or an uncrossed book situation. A crossed book is identified to the user by means of an auction clearing price (*MDEntryType*=Q) (aka indicative or potential auction price). An uncrossed book is identified by means of ToB prices (*MDEntryType* 0=Bid and/or 1=Offer). The visibility of the order book is limited during an auction. Depth information will be explicitly provided again when transitioning from an auction to continuous trading as the user cannot know how much of an order book situation prior to the auction is still valid. Depth information will also be explicitly removed when transitioning from continuous trading to an auction.
- A state transition to Freeze is sent as an *instrument state change* message and does not require any implicit action.
- If the book is crossed, an indicative auction price is calculated and disseminated. The new indicative auction prices are always sent with update action “New”.
- Intraday expired instrument information is provided by a *depth incremental* and *instrument state change* message.
- Only the snapshot and incremental messages of the T7 MDI carry a common and contiguous sequence number per product. The incremental message of T7 EMDI contains a contiguous sequence number per product across all messages, while the snapshot message provides the last sequence number (*LastMsgSeqNumProcessed*) sent in the incremental message.
- Only the best implied price is published. The best implied price will be included in market data only in case it is equal to or better than the best direct price in the respective instrument.
- Whenever the quantity or price of the Best Market changes it is disseminated with update action “New” on the incremental feed. Similarly, the Best Market is removed with update action “Delete”.

Note: The order book is only valid after the entire incremental message has been fully processed.

Figure 13 illustrates a typical order book and terminology used in the following chapters.

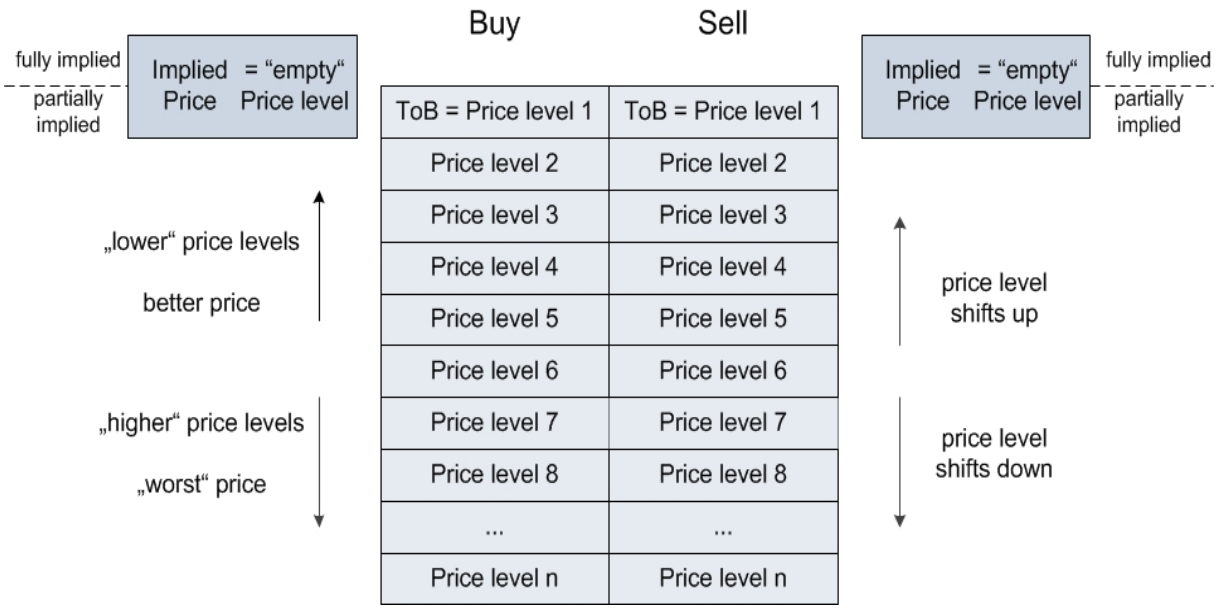

Figure 13: *Typical order book*

An implied price can be either better (fully implied) or the same (partially implied) as price level 1.

9.3.1 Determination of the price sources

T7 supports synthetic matching, where the implied prices from complex instruments can create prices equal or better than the best outright price in the instrument. The implied prices are disseminated in the market data in addition to the prices from outright orders. These prices are shown without a price level. The reported quantities for implied prices and level 1 are not aggregated, i.e. quantities on level 1 are fully outright and do not contain any implied components.

T7 publishes implied prices in market data only in case it is equal to or better than the best outright price in the respective instrument.

In order to find out which situation applies, a price comparison between the implied price (with empty price level) and level 1 (see figure 13) needs to be done:

1. Implied price is better than the outright price at level one -> Fully Implied.
2. Implied price disseminated is equal to the outright price at level 1 -> Partially Implied.
3. Implied price is deleted or absent -> the Best Market price is fully outright and is the same as on level 1.

Examples for all three cases are provided in section 14.2, Example for determination of the price source.

9.3.2 Top Of Book

Simple instruments that are legs of IPS instruments (enabled for synthetic matching) may have synthetic matching opportunities that involve IPS instruments. The corresponding synthetic prices are published with the help of a new Top Of Book Implied data message. Synthetic prices on IPS instruments resulting from their leg instruments are also published via this new Top Of Book Implied data message.

Note: The PriceDepth message continues to contain the order book depth for direct matching and the top-of-book synthetic price derived from synthetic futures spread matching opportunities.

There are two types of synthetic prices due to IPS related matching opportunities that are distributed in the TopOfBookImplied data message:

1. Synthetic prices resulting from synthetic IPS matching opportunities that have no quantity restriction. This price reflects matching opportunities stemming from those IPS instruments that have a leg ratio of 1 in the leg instrument, for which the synthetic price is calculated. Note that the leg ratio condition applies only to that leg. The leg ratios for the other legs of the IPS instrument may have any value.
2. Synthetic prices resulting from synthetic IPS matching opportunities that do have a quantity restriction. This price reflects matching opportunities stemming from those IPS instruments that have a leg ratio greater than 1 in the leg instrument, for which the synthetic price is calculated.

Example: Publish synthetic Top-of-Book without quantity restriction. Bid, Price 106, quantity10.

Tag number	Tag name	Value	Description
35	MsgType	X	MarketDataIncrementalRefresh
34	MsgSeqNum	1069	
49	SenderCompID	75	Unique id of a sender
1300	MarketSegmentID	89	Product
268	NoMDEntries	1	
279	> MDUpdateAction	0	New
269	> MDEntryType	**0**	Bid
1021	> MDBookType	1	Indicates the book type. Always Top-of-Book = 1.
1173	> MDSubBookType	**1**	Indicates the IPS Implied Volume restriction. 1: Implied volume without quantity restriction. 2: Implied volume with quantity restriction.
48	> SecurityID	8852	Instrument
22	> SecurityIDSource	M	Marketplace-assigned identifier
270	> MDEntryPx	**106**	Price
271	> MDEntrySize	**10**	Quantity
273	> MDEntryTime	t_0	official time of book entry
276	> Quotecondition	F	The quote condition. Set to F if crossed. Otherwise empty.

Table 17: *Top Of Book Implied without quantity restriction*

9.3.3 New price level

When a new price level is created in the order book, a *depth incremental* message is sent with field *MDUpdateAction (279)* = 0 ("New"). This indicates that:

- The new price level is to be inserted at the specified price level. [17].
- All existing rows in the order book at the specified and higher levels are to be incremented accordingly. [18].
- Price levels exceeding the maximum specified depth must not be kept in memory.

Note: The field *MDPriceLevel (1023)* is used to identify which level is being inserted.

Example: Buy Limit Order, 10@58.22, enters an empty order book:

Tag number	Tag name	Value	Description
35	MsgType	X	MarketDataIncrementalRefresh
34	MsgSeqNum	1068	
49	SenderCompID	75	Unique id of a sender
1300	MarketSegmentID	89	Product
268	NoMDEntries	1	
279	> MDUpdateAction	0	New
269	> MDEntryType	0	Bid
48	> SecurityID	8852	Instrument
22	> SecurityIDSource	M	Marketplace-assigned identifier
270	> MDEntryPx	**58.22**	Price
271	> MDEntrySize	**10**	Quantity
346	> NumberOfOrders	1	Number of order/quotes on this level
1023	> MDPriceLevel	1	Book level
273	> MDEntryTime	t_0	official time of book entry

Table 18: *MDUpdateAction "New"*

[17] A *MDUpdateAction* (279) = 0 ("New") is also disseminated whenever the quantity changes for the implied price (empty price level).
[18] This is not the case if the *MDUpdateAction* (279) = 0 ("New") is sent for the implied price (with empty price level).

9.3.4 Change of a price level

A *depth incremental* message with *MDUpdateAction* = 1 ("Change") indicates

- A change at a given price level.
- All fields but the price on the specified side at the price level should be updated.

Note: *MDUpdateAction*="Change" is sent only for depth ≥ 1 when the price does not change. A *MDUpdateAction* (279) "Change" contains a price which can be used as a consistency check. However, it never contains a price that is different from the existing one on the current price level.

Example: Quantity changed to 8 for limit order above:

Tag number	Tag name	Value	Description
35	MsgType	X	MarketDataIncrementalRefresh
34	MsgSeqNum	1069	
49	SenderCompID	75	Unique id of a sender
1300	MarketSegmentID	89	Product
268	NoMDEntries	1	
279	> MDUpdateAction	1	Change
269	> MDEntryType	0	Bid
48	> SecurityID	8852	Instrument
22	> SecurityIDSource	M	Marketplace-assigned identifier
270	> MDEntryPx	**58.22**	Price
271	> MDEntrySize	**8**	Quantity
346	> NumberOfOrders	1	Number of order/quotes on this level
1023	> MDPriceLevel	1	Book level
273	> MDEntryTime	t_1	official time of book entry

Table 19: *MDUpdateAction "Change"*

9.3.5 Overlay

A *depth incremental* message with *MDUpdateAction (279)* = 5 ("Overlay") is used to

- Change the price of a given price level. Other parameters, e.g quantity might also change.

Note: *MDUpdateAction*="Overlay" is sent only for depth ≥ 1, i.e. the field *MDPriceLevel (1023)* must be present. In contrast to the *MDUpdateAction*="Change" this instruction contains a price change.

Example: Buy limit order replaces the best buy limit order during instrument state "Auction":

Tag number	Tag name	Value	Description
35	MsgType	X	MarketDataIncrementalRefresh
34	MsgSeqNum	205	
49	SenderCompID	75	Unique id of a sender
1300	MarketSegmentID	70	Product
268	NoMDEntries	1	
279	> MDUpdateAction	5	
269	> MDEntryType	0	Bid
48	> SecurityID	63743	Instrument
22	> SecurityIDSource	M	Marketplace-assigned identifier
270	> MDEntryPx	**2.48**	Price
271	> MDEntrySize	N/A	Quantity remains the same in this example
1023	> MDPriceLevel	1	Book level
273	> MDEntryTime	t_5	official time of book entry

Table 20: *MDUpdateAction "Overlay"*

9.3.6 Deletion of a price level

A *depth incremental* message with *MDUpdateAction (279)*= 2 ("Delete") is used

- to delete a specified price level.

Note: All price levels greater than the deleted one should be decremented. Price and quantity of the price level to be deleted is also sent within the message and can be used as a consistency check.

Example: Deletion of limit order modified above:

Tag number	Tag name	Value	Description
35	MsgType	X	MarketDataIncrementalRefresh
34	MsgSeqNum	1070	
49	SenderCompID	75	Unique id of a sender
1300	MarketSegmentID	89	Product
268	NoMDEntries	1	
279	> MDUpdateAction	2	Delete
269	> MDEntryType	0	Bid
48	> SecurityID	8852	Instrument
22	> SecurityIDSource	M	Marketplace-assigned identifier
270	> MDEntryPx	**58.22**	Price
271	> MDEntrySize	**8**	Quantity
1023	> MDPriceLevel	1	Book level
273	> MDEntryTime	t_2	official time of book entry

Table 21: *MDUpdateAction "Delete"*

9.3.7 Deletion of multiple price levels from a given price level onwards

A *depth incremental* message with *MDUpdateAction (279)* = 4 ("Delete From") is used to

- Delete all price levels $\geq$ specified price level.

Note: All price levels from the specified one and up to the maximum need to be deleted.

Example: Deletion of all orders for SecurityID = 8852, MarketSegmentID = 89 from level 3 and above:

Tag number	Tag name	Value	Description
35	MsgType	X	MarketDataIncrementalRefresh
34	MsgSeqNum	1068	
49	SenderCompID	75	Unique id of a sender
1300	MarketSegmentID	89	Product
268	NoMDEntries	1	
279	> MDUpdateAction	4	Delete From
269	> MDEntryType	0	Bid
48	> SecurityID	8852	Identifier assigned to each instrument
22	> SecurityIDSource	M	Marketplace-assigned identifier
270	> MDEntryPx	**58.19**	Price
271	> MDEntrySize	**13**	Quantity
1023	> MDPriceLevel	**3**	Book level
273	> MDEntryTime	t_3	official time of book entry

Table 22: *MDUpdateAction "Delete From"*

9.3.8 Deletion of multiple price levels up to a given price level

A *depth incremental* message with *MDUpdateAction (279)* = 3 ("Delete Thru") is used to

- Delete all price levels from 1 to the specified price level.

Note: All higher than the specified price levels are shifted down to fill the gap of the deleted price levels.
Example: Deletion of all price levels from 1 to price level 3.

Tag number	Tag name	Value	Description
35	MsgType	X	MarketDataIncrementalRefresh
34	MsgSeqNum	1068	
49	SenderCompID	75	Unique id of a sender
1300	MarketSegmentID	89	Product
268	NoMDEntries	1	
279	> MDUpdateAction	3	Delete Thru
269	> MDEntryType	0	Bid
48	> SecurityID	8852	Unique identifier assigned to each instrument
22	> SecurityIDSource	M	Marketplace-assigned identifier
270	> MDEntryPx	**58.22**	Price on level 3
271	> MDEntrySize	10	Quantity
346	> NumberOfOrders	1	Number of order/quotes on this level
1023	> MDPriceLevel	**3**	Book level
273	> MDEntryTime	t_4	official time of book entry

Table 23: *MDUpdateAction "Delete Thru"*

9.4 T7 Entry Service (TES) Trades

In addition to on-exchange trades T7 EMDI reports ratified disclosed TES trades. For TES trades the *MDOriginType (1024)* is set to 1 = Off-Book. For all other trades the *MDOriginType (1024)* is set to 0 = Book.

An entry consists of

1. *MDOriginType (1024)* is set to 1 = Off-Book.
2. *TradeCondition (277)* field is set to U = Exchange last.
3. *MDEntrySize (271)* and *MDEntryPx (270)* is filled with quantity and price of the trade.
4. *MDEntryID (278)* and the *MDEntryTime (273)*.
5. *MultiLegReportingType (442)* ist set to 1 = Single Security or 3 = Multi Leg Security
6. *MultiLegPriceModel (28750)* is not set.
7. *TrdType (828)* is filled with 1 = BlockTrade, 2 = Exchange for Physical (EFP), 12 = Exchange for Swap (EFS), 54 = OTC, 55 = Exchange Basis Facility, 1000 = Vola Trade, 1001 = EFP-Fin Trade, 1002 = EFP-Index-Futures Trade, 1004 = Block Trade at Market or 1007 = Block QTPIP Trade.

or

1. *MDOriginType (1024)* is set to 1 = Off-Book.
2. *TradeCondition (277)* field is set to U = Exchange last.
3. *MDEntrySize (271)* is filled with quantity of the changed volume.
4. *MDEntryID (278)* and the *MDEntryTime (273)*.
5. *MultiLegReportingType (442)* ist set to 2 = Individual Leg Of A Multi Leg Security
6. *MultiLegPriceModel (28750)* is set to 0 = Standard or 1 = User Defined.
7. *TrdType (828)* is filled with 1 = BlockTrade

When the *TESTradSesStatus (25044)* switches to 5 = PreClose the total *NonDisclosedTradeVolume (28873)* is published. An entry consits of

1. *MDOriginType (1024)* is set to 1 = Off-Book.
2. *MDEntryType (269)* is set to B = Trade Volume
3. *MDEntrySize (271)* is filled with the total quantity of the disclosed TES trades.
4. *NonDisclosedTradeVolume (28873)* is filled with the total quantity of the nonDisclosed TES trades.

The trade statistics of TES trades consists of the trading volume and the last price, it does not include daily high and daily low prices. Consequently, only the trading volume and the last price can be recovered.

9.5 Manual Trade Entry and Trade Reversal (T7 EMDI)

The T7 EMDI reports all on-exchange trades executed on T7. In addition to order book trades, members receive trade messages for trades or trade reversals manually entered by Market Supervision.

The following fields are not sent for trade entries and trade reversals: *AggressorSide (2446)*, *AggressorTime (2445)*, *RequestTime (5979)*, *NumberOfBuyOrders (2449)*, *NumberOfSellOrders (2450)*.

The T7 MDI does not report manual trade entries nor trade reversals as only statistical information is provided.

9.5.1 Manual Trade Entry (by Market Supervision) (T7 EMDI)

The entry consists of

1. *TradeCondition (277)* field are always set to k = Out Of Sequence.
2. *MDEntryType (269)* field is always set to 2 = Trade.
3. *MDEntrySize (271)* and *MDEntryPx (270)* is filled with quantity and price of the trade.
4. *MDEntryID (278)* and the *MDEntryTime (273)*.

A manually entered trade will not affect the price statistics. Even when the manually entered trade is higher than the daily high price, it does not change the daily high price. For that reason the field *TradeCondition (277)* for a manually entered trade must only contain the “Out Of Sequence” attribute.

9.5.2 Trade Reversal (by Market Supervision) (T7 EMDI)

A trade reversal is triggered by Market Supervision in order to delete a trade completely. A trade can only be reversed with its complete quantity.

Deleting a trade may affect the Trade Volume Report. Sometimes one or more price statistics are adjusted. An incremental for a trade reversal consists of one entry with *MDUpdateAction* 2 = Delete and potentially one or more entries with *MDUpdateAction* 1 = Change per involved instrument.

The entry with *MDUpdateAction* 2 = Delete consists of

- *MDEntrySize (271)* and *MDEntryPx (270)* of the reversed trade.
- *MDEntryType (269)* is set to 2 = Trade.
- *MDEntryID (278)* (match event identifier) of the reversed trade.
- *MDEntryTime (273)* is set to the entry time of the reversed trade.

Incremental entries with *MDUpdateAction* 1 = Change provide information about what was affected by the reversal. Each entry consists of

- *MDEntrySize (271)* and *MDEntryPx (270)* if a new last price is set.
- *MDEntryPx (270)* the new price of the affected price statistics (e.g. High, Low, Opening, Closing).
- *MDEntryType (269)* is set to 2 = Trade.
- *TradeCondition (277)* contains the related set of trade conditions for the affected price statistics.
- *MDEntryTime* (273) of the updated last trade if *TradeCondition* (277) contains “Exchange Last”.

9.6 Trade Volume Reporting (T7 EMDI)

All on-exchange trades executed on T7 are reported via *depth incremental* messages. The *depth snapshot* messages contain statistical information about trades only. Trades can be identified in the incremental messages when *MDEntryType* is set to 2 (Trade).

The T7 EMDI disseminates information about book and Off-Book trades. The *MDOriginType (1024)* is set to Book or Off-Book accordingly.

When an order executes against the book at multiple price levels, this is reflected by a matching event with multiple match steps. Each match step has the trades at one price level and is represented by a unique *MDEntryID (278)* and published in the market data.

The field *MDEntryID (278)* is a unique id on product level and origin type for each business day.

9.6.1 Use case 1: Direct match of simple instruments

An incoming simple order is matched against two orders of the opposite side of the order book on different price levels.

Incoming buy order, 10@85, BMW

Existing Order book:

Bid	Ask
	5@84.9
	5@85

Trade Volume Reporting: Two trades are reported because two different price levels are involved in the matching process: First 5@84.9 gets reported due to a higher matching priority of this price level; afterwards 5@85.

Instr.	MDEntryID	MDUpdateAction	size@prc	TradeCond.	AggrSide	#Buy	#Sell
BMW	1	NEW	5@84.9	U,R,AX,AY	BUY	1	1
BMW	2	NEW	5@85	U,AX	BUY	1	1

with:
U = Exchange last
R = Opening price
AX = High price
AY = Low price
AW = Last auction price

9.6.2 Use case 2: Self-Match prevention (order is totally cancelled)

An incoming order is cancelled due to Self-Match prevention.

Incoming buy order, 150@84, FESX Mar, MatchInstCrossID=1, Member A

Existing Order book:

Bid	Ask
	50@84 (MatchInstCrossID=1, Member A)

Trade Volume Reporting: A trade is reported: 0@84, AggressorSide BUY. MDEntryID, TradeCondition, number of Buy and number of Sell orders are not filled. The resting cancelled quantity is 50. The incoming cancelled quantity (150) is not reported.

Instr.	MDEntryID	MDUpdateAction	size@prc	TradeCond.	AggrSide	#Buy	#Sell	#RestingCxlQty
BMW		NEW	0@84		BUY			50

9.6.3 Use case 3: Self-Match prevention (order is partially cancelled)

An incoming order is partially cancelled due to Self-Match prevention.

Incoming buy order, 150@84, BMW, MatchInstCrossID=1, Member A

Existing Order book:

Bid	Ask
	20@84 (MatchInstCrossID=1, Member A)
	30@84 (MatchInstCrossID=0)

Trade Volume Reporting: A trade is reported: 30@84, AggressorSide BUY. The resting cancelled quantity is 20.The incoming cancelled quantity (120) is not reported.

Instr.	MDEntryID	MDUpdateAction	size@prc	TradeCond.	AggrSide	#Buy	#Sell	#RestingCxlQty
BMW	1	NEW	30@84	U	BUY	1	1	20

9.6.4 Use case 4: Opening auction

After the uncrossing of the order book in a simple instrument at the end of an auction call phase, five orders on the buy side and 3 orders on the sell side of the order book have been matched. The *Trade-Condition (277)* is set to AW for Auctions. The field *TrdType (828)* specifies the type of the auction. For on-exchange trades outside the auction, *TrdType* (828) is not set.

Existing Order Book during Auction:

Bid	Ask
30@24.39 Sep	60@24.39 Sep
25@24.39 Sep	57@24.39 Sep
20@24.39 Sep	18@24.39 Sep
55@24.39 Sep	
5@24.39 Sep	

Trade Volume Reporting: All orders are matching on the same price level. Therefore they are reported only once but with different *NumberOfBuyOrders (2449)* /*NumberOfSellOrders (2450)* . The *Aggressor-Side (2446)* is left empty because during an auction, orders are not considered to be aggressive.

The following *depth incremental* message is sent:

Instr.	MD-EntryID	MDUpdate-Action	size@prc	TradeCond.	TrdType	AggrSide	#Buy	#Sell
Sep	1	NEW	135@24.39	U,R,AX,AY,AW	OPENING		5	3

The following *depth snapshots* belong to the *depth incremental* above:

Instr.	MDUpdateAction	size@prc	TradeCondition	TrdType
Sep	NEW	135@24.39	U,R,AX,AY	
Sep	NEW	135@24.39	AW	OPENING

In the snapshot, the last auction prices are published in dedicated entries for each auction type separately. Each additional trade from another auction type, adds an entry in the snapshot up to a maximal number of four entries, one for each type of auction. If an auction trade gets reversed the respective snapshot entry for the auction trade does not get deleted.

9.7 Trade Volume Reporting (T7 EMDI), Cash Only

9.7.1 Reference Price and Price Without Turnover

For cash market instruments a reference price is published during system start. *MDEntryID (278)* is not set. *MDEntrySize (271)* is set to 0. *Trade Condition (277)* is set to U = Exchange Last.
Auctions may result in an auction price without turnover. An auction price without turnover is regarded as a regular auction price, thus updating last and potentially high and low price. *MDEntryID (278)* is not set. *MDEntrySize (271)* is set to 0.

Please note, that totally cancelled trades resulting from Self-Match prevention are also reported with *MDEntrySize (271)* set to 0 and *MDEntryID (278)* not set, but *RestingCxlQty (28869)* will be greater 0 (see 9.8.2).

9.7.2 Use case 1: Algorithmic Trade Indicator

The field *AlgorithmicTradeIndicator (2667)* indicates an Algorithmic Trade, i.e. at least one matching order was submitted by a *trading algorithm* instead of a *human* being. This flag is not used in derivative markets.

An incoming simple order is matched against two orders of the opposite side of the order book on different price levels.

Incoming buy order, 3@97.32, DB1 (*human*)

Existing Order book (DB1):

Bid	Ask
	1@97.31 (*human*)
	1@97.32 (*human*)
	1@97.32 (*trading algorithm*)

Trade Volume Reporting: Two trades are reported because two different price levels are involved in the matching process: A first trade 1@97.31 is reported with *AlgorithmicTradeIndicator (2667)* not set since no order from a *trading algorithm* is involved. A second trade 2@97.31 is reported with *AlgorithmicTradeIndicator (2667)* set to 1 = Algorithmic Trade since an order from a *trading algorithm* is involved.

Instr.	MDEntryID	MDUpdateAction	size@prc	TradeCond.	AlgoInd.	AggrSide	#Buy	#Sell
DB1	10	NEW	1@97.31	U		BUY	1	1
DB1	11	NEW	2@97.32	U	1	BUY	1	2

9.8 Trade Volume Reporting (T7 EMDI), Derivatives Only

A synthetic match can result in more than one trade volume record with the same *MDEntryID (278)* as shown in 9.8.1 and 9.8.2.

Every match step occurring in the exchange has an identifier in T7 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). This identifier allows participants to link trade capture reports and the corresponding execution report of the T7 ETI with the market data incremental feed of the T7 EMDI.

In case of a market data feed restart, the *MDEntryID (278)* is set to NULL in each *MDIncGrp* entry of the first *Depth Incremental* message after the *MsgSeqNum (34)* is reset to value 1 (see paragraph 9.10.3, Market data feed restart (T7 EMDI)). Member applications that look at the *TradeCondition (277)* value ”Exchange Last” (=U) should also check whether an *MDEntryID (278)* is set before they use the

MDEntrySize (271) to derive a new trade volume from the previous one. If *MDEntryID (278)* is absent then the trade provides the last valid trade prior to the restart and not a new trade after the restart.

The *AggressorTime (2445)* and *RequestTime (5979)* timestamps are provided for the incoming orders when they lead to an immediate execution. In some cases they are not published, for example for trades resulting from an auction uncrossing. It is also possible that the *AggressorSide (2446)* appears without *AggressorTime (2445)* information.

The *RestingCxlQty (28869)* is provided when a resting order is deleted due to a Self-Match prevention (SMP) event. There may be a SMP event in the context of a trade. But there could also occur a pure SMP event. In this case, there is no *MDEntryID (278)* and the *MDEntrySize (271)* is zero.

The traded size on simple instruments not involving any simple instrument orders (e.g. direct match of complex instruments) is published via an additional *depth incremental* message having *Trade Condition (277)* set to 'a' (Volume only).

The following use cases illustrate the *MDEntryID (278)* and how Trade Volume Reporting works. Please note that, the leg ratio is assumed always 1 for the following uses cases.

9.8.1 Use case 1: Complex versus simple order match

A buy spread order as an incoming complex order (Time Spread) matches (synthetically) against several simple instrument leg orders (outright orders).

Incoming buy order, 200@8.0 FESX Sep/Dec

Existing Order book:

Bid	Ask
120@2878 Dec	
30@2878 Dec	

Bid	Ask
	60@2886.0 Sep
	50@2886.0 Sep
	40@2886.0 Sep

This results in the following implied price:

Bid	Ask
	Sep12/Dec12 150@8.0

Trade Volume Reporting: The incoming spread order matches against the implied-in order of the order book which is a composition of all 5 outright orders in the order book. Again, the trades are aggregated per price level. The fields *NumberOfBuyOrders (2449)* and *NumberOfSellOrders (2450)* show how many orders are involved.

In case of a synthetically matched complex order either the buy or sell side contains an empty value. In case of a direct matched complex instrument, both sides are filled.

Instr.	MDEntryID	MDUpdateAction	size@prc	TradeCond.	AggrSide	#Buy	#Sell
Sep/Dec	5	NEW	150@8	U	BUY	1	
Sep	5	NEW	150@2886	U,AX			3
Dec	5	NEW	150@2878	U,R,AX,AY		2	

9.8.2 Use case 2: Complex versus simple/complex match

Incoming buy order, 250@8, FESX Sep/Dec

Existing Order book:

Bid	Ask
120@2878 Dec 30@2878 Dec	

Bid	Ask
	150@2886 Sep

Bid	Ask
	100@8 Sep/Dec

Trade Volume Reporting: Incoming complex order is matching directly against the opposite side of a complex order; another part is matching against an implied-in order which was created by two existing outright orders for the Sep and Dec contracts. The direct match of the complex orders can be identified by existing entries for *NumberOfBuyOrders (2449)*, *NumberOfSellOrders (2450)*. The synthetic match can be identified by the missing entry for *NumberOfSellOrders (2450)*.

Instr.	MDEntryID	MDUpdateAction	size@prc	TradeCond.	AggrSide	#Buy	#Sell
Sep/Dec	6	NEW	100@8	U	BUY	1	1
Sep/Dec	6	NEW	150@8	U	BUY	1	
Sep	6	NEW	150@2886	U			1
Dec	6	NEW	150@2878	U,AY		2	
Sep	6	NEW	100@-	a			
Dec	6	NEW	100@-	a			

9.9 Trade Volume Reporting (T7 MDI)

The T7 MDI only provides statistical data (daily high/low price as well as total trade volume) for trades as well as the last traded price and quantity. Other information such as *NumberOfBuyOrders (2449)*, *NumberOfSellOrders (2450)* are not provided.

For each simple instrument participating in a trade, T7 MDI reports the total traded volume even when there are no simple instrument orders involved in the trade (e.g. direct match of complex instruments).

9.10 Failure of the market data feed / matching engine

The following chapters explain fail-over scenarios and how receiving applications need to process them.

9.10.1 Normal processing

At start-up, the system assigns a unique sender identifier, the *SenderCompID (49)* to each market data feed. Afterwards the *SenderCompID (49)* remains constant for a given product during the entire business day. The *SenderCompID (49)* as shown in section 7.1 is available in the *packet header* and in the data message[19], e.g. *depth incremental* or *depth snapshot* itself.

For each incremental and snapshot message sent by market and reference data feeds:

- the field content for *SenderCompID (49)* in the *packet header* and in each data message is always the same.

For each incremental and snapshot message sent by the market data feeds:

- the *PacketSeqNum's* in the *packet header* are contiguous per *SenderCompID*, multicast address and port combination.
- the *MsgSeqNum's* in the data message are contiguous per product on the incremental feed of the T7 EMDI.
- the *MsgSeqNum's* in the data message are contiguous per product on the market data feed of the T7 MDI[20].

Figure 14 provides an example for constant *SenderCompID's* and increasing sequence numbers:

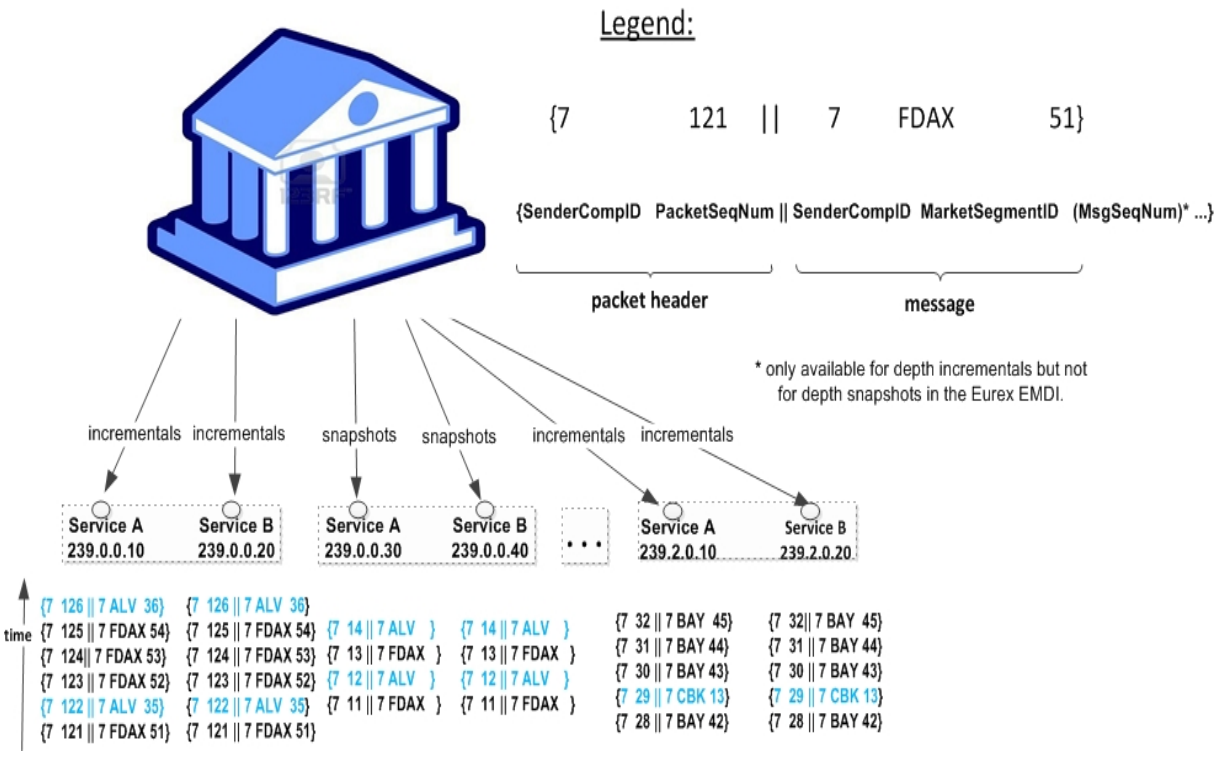

Figure 14: *Normal processing of sequence numbers for the T7 EMDI*

[19] the content is the same.
[20] because the T7 MDI delivers incrementals and snapshots on the same channel.

9.10.2 Market data feed fail-over (T7 MDI)

A new *SenderCompID*, available in the *packet header* and in each data message for incrementals and snapshots, indicates a fail-over of the T7 MDI market data feed. During a fail over applications may receive the old and the new *SenderCompID* simultaneously for a period of time. Therefore the old *SenderCompID* needs to be ignored on that specific channel for the rest of the business day and only the new *SenderCompID* should be processed further on. However, the old *SenderCompID* might be re-used on the next business day again.

Incrementals and Snapshots:

- the *PacketSeqNum's* in the *packet header* are reset to 1 and are contiguous per *SenderCompID (49)*, multicast address and port combination.
- the *MsgSegNum's* for effected products are reset to 1. This applies to incrementals and snapshots as both are sent inline on the same channel (multicast address and port combination).

Figure 15 illustrates the behaviour for incremental and snapshot messages:

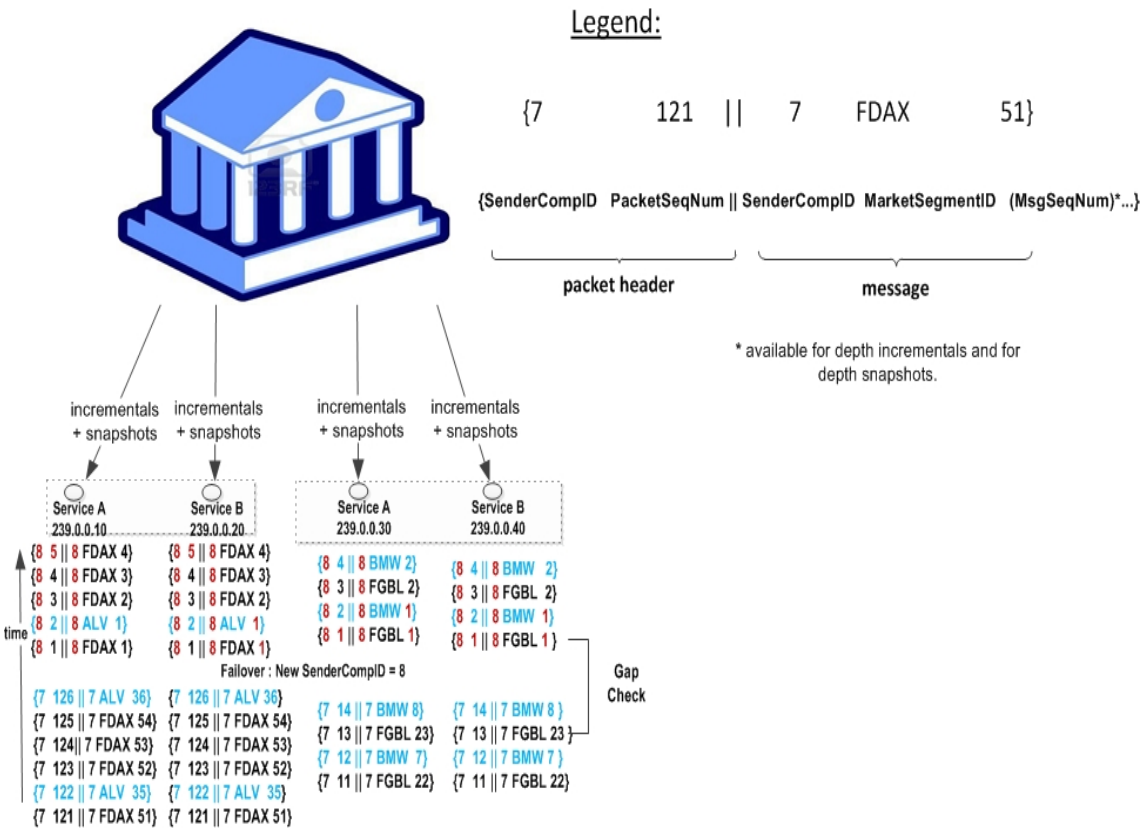

Figure 15: *Data feed fail-over for T7 MDI*

Participants can identify this failover scenario by decoding the *packet header* of a UDP datagram and comparing the *SenderCompID* value with the previous value.

Please note that EMDI, MDI and RDI interfaces each have their own private range of numbers for the *SenderCompID*'s. Therefore EMDI, MDI and RDI might use the same *SenderCompID* and applications need to check to which channel the *SenderCompID* belongs to.

In case an application starts synchronization right in the middle of a fail-over period, it could happen, that an application might start synchronization on the *new SenderCompID*. If the application now receives a

packet with the *old* one, it will switch to the *old* and further ignoring the *new* one. In this specific case an application should run into a timeout after a period of time, when no packets with the *old SenderCompID* are received anymore. The application needs to ignore the *old SenderCompID* further on, remove the *new SenderCompID* from the dropped list and restart synchronization again.

Note: In general, channels and their associated multicast addresses could be shared among different senders, e.g. all Xetra (XETR) DAX products could be disseminated on one channel, but will originate from more than one sender. Therefore, client applications must be prepared to maintain the *PacketSeqNum* per channel / *SenderCompID* combination and a fail-over or restart of a market data sender could only be detected reliably, if a change of *SenderCompID* is detected for a specific product as depicted in Figure 15.

Note: A *new SenderCompID* could be any number less or equal to 127, that was not used before on a specific channel on a specific business day.

9.10.3 Market data feed restart (T7 EMDI)

A new *SenderCompID*, available in the *packet header* and in each data message for incrementals and snapshots, indicates a failure.

Incrementals:

- the *PacketSeqNum's* in the *packet header* are reset to 1 and are contiguous per *SenderCompID*, multicast address and port combination.
- the *MsgSeqNum's* in the data message are reset to 1 and are contiguous per product for incrementals. There is also a *Depth Incremental* message sent on the incremental feed that contains a full refresh of the Trade Statistics equivalent to the Trade Statistics that are also sent on the snapshot feed after the restart. Therefore member applications do not need to listen to the snapshot feed for synchronizing incrementals and snapshots. Please note that these Trade Statistics do not contain any *MDEntryID (278)* as for a regular trade event. See paragraph 9.6, Trade Volume Reporting (T7 EMDI).

Snapshots:

- the *PacketSeqNum's* in the *packet header* are reset to 1 and are contiguous per *SenderCompID*, multicast address and port combination.

Once this condition is observed it is safe to assume that a failure scenario took place and the only correct action is to rebuild the order book. The receiving application needs to invalidate its view of the order book until an explicit message has been received containing new information. This can either be as a result of a recovery from *depth snapshots* or from *depth incremental* messages, as described in section 6.4.1, Build the initial order book with the T7 EMDI.

Note: A *new SenderCompID* could be any number less or equal to 16383, that was not used before on a specific channel on a specific business day.

9.10.4 Market data feed restart (T7 MDI)

Receiving applications are able to identify a failure as follows:

- by a change of the *SenderCompID (49)* in the *packet header* and in all subsequent messages.
- by a reset of the *MsgSegNum's* for all products to 1.

Once this condition is observed it is safe to assume that a failure scenario took place and the only correct action is to rebuild the order book. The receiving application needs to invalidate its view of the order book until an explicit message has been received containing new information. This can either be as a result of a recovery from *depth snapshots* or from *depth incremental* messages, as described in section 6.4.2, Build the initial order book with the T7 MDI.

9.10.5 Failure of the matching engine

All non-persistent orders and quotes are deleted. Participants can see a product state change as a result of the market reset. The same rules apply as for 9.10.3 Market data feed restart (T7 EMDI).

In addition, participants receive a market reset event from their ETI-interface. The *service availability* message indicates the unavailability of the matcher by the ETI-field *MatchingEngineStatus (25005)*.

9.11 Trading states for a sample business day for derivates

Section 4.2, Trading states introduced the trading state information. This section describes a typical day with T7. The example refers to the FDAX future on an expiry day. The times for each trading phase are valid for FDAX.

Participants should not rely on any specific order or sequence of messages as described in the following chapters. For instance, the system could send an *instrument state change* message instead of a *mass instrument state change* message resulting in the same trading state at the participants' side.

Unless participants rely on the message-specific fields (TradingSessionID (336), TradingSessionSubID (625) and TradSesStatus (340)), the product state change messages don't have to be processed in order to receive the correct order book state; it is sufficient to process the *instrument state* and *mass instrument state* change messages.

9.11.1 Start-Of-Day

The system startup occurs in the morning. Note that with T7, business days are not technically linked to the local calendar. Under normal circumstances a business date is equal to the local calendar date. Nevertheless it is possible that the system startup and with it the new business day starts before midnight on the previous calendar day.

At startup, the FDAX product goes into the product state "Start-of-Day", while all its instruments are in the state Closed. Traders have no access to the order book.

The system sends a *product state change* message (FIX *TradingSessionStatus* (MsgType = h)) with the field *TradingSessionID (336)* set to 3 = Morning and the field *TradingSessionSubID (625)* set to 7 = Quiescent. This indicates the product state "Start-of-Day".

The system furthermore sends *mass instrument state change* message (FIX *SecurityMassStatus* (MsgType = CO)) with the field *SecurityMassTradingStatus (1679)* containing 200 = Closed, which indicates that all instruments are in the state Closed. This message is sent once for the futures contracts (specified in the field *InstrumentScopeProductComplex (1544)* containing 1 = Simple Instrument) and once for futures spreads (specified in the field *InstrumentScopeProductComplex (1544)* containing 5 = Futures Spread) which is the only complex instrument type supported for futures.

The reference data feed begins with the system startup. Instruments that are scheduled to expire during the day are included in the reference data, but instruments that have already expired on a previous business day are no longer included in the reference data.

9.11.2 Pre-Trading

At **7:30 CET**, the FDAX product goes into the product state Pre-Trading while all its instruments change their instrument state to Book. Traders are now able to perform full order and quote maintenance.

The system sends a *product state change* message with the field *TradingSessionID (336)* set to 3 = Morning and the field *TradingSessionSubID (625)* set to 1 = Pre-Trading. This indicates the product state Pre-Trading.

The system furthermore sends *mass instrument state change* message with the field *SecurityMassTradingStatus (1679)* containing 202 = Book, which says that all instruments are in the state Book. This message is sent once for simple instruments and once for futures spreads.

9.11.3 Opening Auction

At **7:50 CET**, the FDAX product goes into the product state Trading. At the same time, all its simple instruments (futures contracts) change their instrument state to Opening Auction. The complex instruments (futures spreads) remain in the instrument state Book. Traders can do full order and quote maintenance.

For the simple instruments, the system publishes the best bid and ask prices if the order book is not crossed, or an indicative auction price if the order book is crossed.

The system sends a *product state change* message with the field *TradingSessionID (336)* set to 1 = Day and the field *TradingSessionSubID (625)* set to 3 = Continuous. This indicates the product state Trading.

The system furthermore sends one *mass instrument state change* message with the field *SecurityMassTradingStatus (1679)* containing 204 = Opening Auction, which says that all instruments are in the state opening auction. This message is sent only for simple instruments. There is no message sent for futures spreads as they do not change their state.

9.11.4 Continuous Trading

At **8:00 CET**, the opening auction period of the FDAX product ends and continuous trading starts. There is no product state change involved, but all the instruments transition to the instrument state Continuous. The change of the instrument state implies an auction trade if the order book was crossed. This applies also to the complex instruments (futures spreads), even though they had no formal auction call phase before.

In the instrument state Continuous, traders can maintain their orders and quotes. Incoming orders and quotes are continuously matched. The system publishes order book and trade data.

The system sends two *mass instrument state change* messages with the field *SecurityMassTradingStatus (1679)* containing 203 = Continuous, which means that all instruments are in the state Continuous. This message is sent once for simple instruments and once for futures spreads.

9.11.5 Intraday Expiry

At **13:00 CET**, the front month contract of the FDAX future expires on an expiration day. The affected simple instrument goes to the instrument state Restricted. The same happens to all complex instruments (futures spreads) that have the affected simple instrument as a leg. For these instruments, all quotes are deleted automatically. Traders may delete their orders but not modify them or add new orders.

For the expired simple instrument, the system sends a *instrument state change* message (FIX *SecurityStatus* (MsgType = f)) with the field *SecurityTradingStatus (326)* containing 201 = Restricted, which says that this particular instrument is in the state Restricted. Furthermore, the field *SecurityStatus (965)* contains the value 4 = Expired.

For each impacted complex instrument, the system sends a *instrument state change* message with the field *SecurityTradingStatus (326)* containing 201 = Restricted and field *SecurityStatus (965)* containing the value 4 = Expired.

9.11.6 Closing Auction

At **22:00 CET**, the FDAX product is set into the product state Closing. At the same time, all its simple instruments (futures contracts) change their instrument state to Closing Auction. The complex instruments (futures spreads) change to the instrument state Book. Traders can do full order and quote maintenance. For simple instruments, the system publishes the best bid and ask prices if the orderbook is not crossed, or an indicative auction price if the order book is crossed.

The expired front month contract and the related futures spread instruments are not affected. They remain in the state Restricted.

The system sends a *product state change* message with the field *TradingSessionID (336)* set to 1 = Day and the field TradingSessionSubID (625) set to 4 = Closing. This indicates the product state Closing.

For simple instruments, the system sends one *mass instrument state change* message with the field *SecurityMassTradingStatus (1679)* containing 210 = Closing Auction. The message carries an exception list which contains the expired instrument as the only list item. For this instrument, the list item field *SecurityTradingStatus (326)* contains 201 = Restricted.

For the futures spreads, the system sends one *mass instrument state change* message with the field *SecurityMassTradingStatus (1679)* containing 202 = Book. The message carries an exception list which contains all the futures spreads that are in the state Restricted. For these instruments, the list item field *SecurityTradingStatus (326)* contains 201 = Restricted.

9.11.7 Post-Trading

At **22:03 CET**, the closing auction period of the FDAX product ends. The product FDAX goes into the product state Post-Trading. The simple instruments that have been in the instrument state Closing Auction now change to the state Book. The other instruments do not change their state.

The expired front month contract and the related futures spread instruments are not affected. They remain in the state Restricted.

For the instruments that are in the instrument state Book, traders can do full order and quote maintenance.

The system sends a *product state change* message with the field *TradingSessionID (336)* set to 5 = Evening and the field *TradingSessionSubID (625)* set to 5 = Post-Trading. This indicates the product state: Post-Trading.

For simple instruments, the system sends one *mass instrument state change* message with the field *SecurityMassTradingStatus (1679)* containing 202 = Book. The message carries an exception list which contains the expired instrument as the only list item. For this instrument, the list item field *SecurityTradingStatus (326)* contains 201 = Restricted.

For the futures spreads, the system sends one *mass instrument state change* message with the field *SecurityMassTradingStatus (1679)* containing 202 = Book. The message carries an exception list which contains all the futures spreads that are in the state Restricted. For these instruments, the list item field *SecurityTradingStatus (326)* contains 201 = Restricted.

9.11.8 End-Of-Day

After **22:30 CET**, the FDAX product goes into the product state End-Of-Day. All its instruments change into the instrument state Closed. Traders can no longer access the order book. The exchange system will start the end-of-day processing.

The system sends a *product state change* message with the field *TradingSessionID (336)* set to 5 = Evening and the field *TradingSessionSubID (625)* set to 7 = Quiescent. This indicates the product state End Of Day

The system also sends two *mass instrument state change* messages with the field *SecurityMassTradingStatus (1679)* containing 200 = Closed, which means that all instruments are in the state Closed. This message is sent once for simple instruments and once for future spreads.

9.12 Trading states for a sample business day for cash instruments

Section 4.2, Trading states introduced the trading state information. This section describes a typical day with T7 for cash instruments. The times for each trading phase are valid for DAX instruments.

While the trading states for cash and derivatives are similar there are some differences: there is typically only one instrument per product (and therefore no mass state change), there is typically an intraday auction and the schedule is different from the derivatives schedule.

Participants should not rely on any specific order or sequence of messages as described in the following chapters. For instance, the system could send an *instrument state change* message instead of a *mass instrument state change* message resulting in the same trading state at the participants' side.

Unless participants rely on the message-specific fields (TradingSessionID (336), TradingSessionSubID (625) and TradSesStatus (340)), the product state change messages don't have to be processed in order to receive the correct order book state; it is sufficient to process the *instrument state* and *mass instrument state* change messages.

9.12.1 Start-Of-Day

The system startup occurs in the morning. Note that with T7, business days are not technically linked to the local calendar. Under normal circumstances a business date is equal to the local calendar date. Nevertheless it is possible that the system startup and with it the new business day starts before midnight on the previous calendar day.

At startup, products go into the product state "Start-of-Day", while all instruments are in the state Closed. Traders have no access to the order book.

The system sends *product state change* messages (FIX *TradingSessionStatus* (MsgType = h)) with the field *TradingSessionID (336)* set to 3 = Morning and the field *TradingSessionSubID (625)* set to 7 = Quiescent. This indicates the product state "Start-of-Day".

The system furthermore sends *instrument state change* messages (FIX *SecurityMassStatus* (MsgType = CO)) with the field *SecurityTradingStatus (326)* containing 200 = Closed, which indicates that instruments are in the state Closed.

The reference data feed begins with the system startup.

9.12.2 Pre-Trading

At **7:30 CET**, the cash products go into the product state Pre-Trading while the instruments change their instrument state to Book. Traders are now able to perform full order and quote maintenance.

The system sends a *product state change* message with the field *TradingSessionID (336)* set to 3 = Morning and the field *TradingSessionSubID (625)* set to 1 = Pre-Trading. This indicates the product state Pre-Trading.

The system furthermore sends *instrument state change* message with the field *SecurityTradingStatus (326)* containing 202 = Book, which says that all instruments are in the state Book.

9.12.3 Opening Auction

At **8:50 CET**, the cash products go into the product state Trading. At the same time, all instruments change their instrument state to Opening Auction. Traders can do full order and quote maintenance. The system publishes the best bid and ask prices and quantities if the order book is not crossed, or an indicative auction price and quantity if the order book is crossed.

The system sends a *product state change* message with the field *TradingSessionID (336)* set to 1 = Day and the field *TradingSessionSubID (625)* set to 3 = Continuous. This indicates the product state Trading.

The system furthermore sends *instrument state change* messages with the field *SecurityTradingStatus (326)* containing 204 = Opening Auction, which says that instruments are in the state opening auction.

9.12.4 Continuous Trading

At **9:00 CET**, the opening auction period of the DAX instruments ends and continuous trading starts. There is no product state change involved, but all the instruments transition to the instrument state Continuous. The change of the instrument state implies an auction trade if the order book was crossed.

In the instrument state Continuous, traders can maintain their orders and quotes. Incoming orders and quotes are continuously matched. The system publishes order book and trade data.

The system sends *state change* messages with the field *SecurityTradingStatus (326)* containing 203 = Continuous, which means that instruments are in the state Continuous.

9.12.5 Intraday Auction

At **1:00 p.m. CET**, intraday auction for DAX instruments starts. The system sends *instrument state change* messages with the field *SecurityTradingStatus (326)* containing 206 = Intraday Auction. Traders can do full order and quote maintenance. The system publishes the best bid and ask prices and quantities if the orderbook is not crossed, or an indicative auction price and quantity if the order book is crossed. After 2 minutes the auction is committed and the system sends *state change* messages with the field *SecurityTradingStatus (326)* containing 203 = Continuous.

9.12.6 Closing Auction

At **5:30 p.m. CET**, the cash products are set to the product state Closing. At the same time, the instruments change their instrument state to Closing Auction. Traders can do full order and quote maintenance. The system publishes the best bid and ask prices and quantities if the orderbook is not crossed, or an indicative auction price and quantity if the order book is crossed.

The system sends a *product state change* message with the field *TradingSessionID (336)* set to 1 = Day and the field TradingSessionSubID (625) set to 4 = Closing. This indicates the product state Closing.

The system sends *instrument state change* message with the field *SecurityTradingStatus (326)* containing 210 = Closing Auction.

9.12.7 Post-Trading

At **5:35 p.m. CET**, the closing auction period of the DAX products end. The products go into the product state Post-Trading. The instruments that have been in the instrument state Closing Auction now change to the state Book.

The system sends a *product state change* message with the field *TradingSessionID (336)* set to 5 = Evening and the field *TradingSessionSubID (625)* set to 5 = Post-Trading. This indicates the product state: Post-Trading.

The system sends *instrument state change* message with the field *SecurityTradingStatus (326)* containing 202 = Book.

9.12.8 End-Of-Day

After **8:30 p.m. CET**, the DAX products go into the product state End-Of-Day. All instruments change into the instrument state Closed. Traders can no longer access the order book. The exchange system will start the end-of-day processing.

The system sends a *product state change* message with the field *TradingSessionID (336)* set to 5 = Evening and the field *TradingSessionSubID (625)* set to 7 = Quiescent. This indicates the product state End Of Day

The system also sends *instrument state change* messages with the field *SecurityTradingStatus (326)* containing 200 = Closed, which means that instruments are in the state Closed.

10 Fine tuning client applications

This chapter covers some aspects of application tuning which should be considered during the design process of receiving applications.

10.1 Buffer size

Messages need to be buffered and sorted in order to deal with datagrams arriving in reversed order. A bigger buffer size usually slows down the processing of messages and should therefore be avoided. Conversely, receiving applications might falsely declare a message as lost if the buffer size is too small. As you can see from this example, a bigger buffer size works contrary to the speed of an application but reduces the chances of “lost” messages.

Another factor which effects the ideal buffer size is the ratio of joined multicast streams to available bandwidth of an T7 Market Data connection. A connection which operates at high network utilization levels might more often cause multicast drops or packets arriving in an incorrect sequence.

Last but not least, the location of the receiving application also matters. For instance, an application running in co-location has very few out-of-order multicast packets (none in most cases) while an application which is located at a far distance from the T7 host receives a few packets out-of-order.

Therefore a general recommendation concerning the buffer size cannot be made. Developers need to determine the ideal buffer size during internal testing. Please take into account that the message rate for the public broadcast is usually much lower in the simulation environment than it is in production.

10.2 Packet and message processing

It is important that messages are removed from the network in a timely fashion to prevent them from being dropped by the client machine due to "receive buffer" overflow in the IP stack. In addition to the removal of messages from the network stack (as might be performed in response to a select() operation, for example), this design requires a time-based component to determine when a missing packet is declared lost (as opposed to simply delayed).

The mechanism behind this is an implementation detail, and is platform-specific, but in its simplest form a timed select() and polling of an internal list of overdue packets would suffice.

The actual time out value applied is very implementation-specific, and may be either determined dynamically (with a knowledge of when the first overdue packet is declared lost) or a simple static value.

Note: *Depth incremental* messages must not be applied to the order book unless they are in sequence.

For each network packet received, decode it into the constituent FIX message and then for each message:

The market data feeds may contain information about multiple products. If it is not for a product that the clients application is interested in:

- Throw it away.

If the message is already in the cache:

- The clients application already received this message from the mirror channel, or it has been duplicated in the network.
- Throw it away.

Otherwise:

- Add it to the cache.

10.3 Application level

Various approaches can lead to faster processing on application level. The approaches depend primarily on the purpose and algorithm of the application.

10.3.1 Discarding duplicate packets within the live-live environment

It is expected that receiving applications process packets from Service A and B simultaneously. The concept of the *packet header* allows receiving applications to detect duplicates based on the *PacketSeqNum*. It is recommended to discard a packet after decoding the *packet header* once it has been identified as duplicate. The actual message following the *packet header* no longer has to be decoded, allowing a faster processing speed.

10.3.2 Order book processing

Depth incremental messages deliver changes of the order book from ToB to worse price levels. Trading algorithms which are based on fast matching without the knowledge of the order book could process ToB only before making a decision and process the order book afterwards.

Conversely, trading algorithms with a matching logic based on the knowledge of the order book need to process the order book before sending orders/quotes.

10.3.3 Optimal processing of desired products (T7 EMDI)

Receiving applications interested in certain products need to join a multicast address which contains the desired products according to the mapping table provided in the reference data. Packets may arrive from different partitions on the same multicast address as shown in figure 16. The PartitionID in the *packet header* for the T7 EMDI can be used to identify packets arriving from partitions which carry the desired products. All other packets can be easily discarded without decoding the entire message.

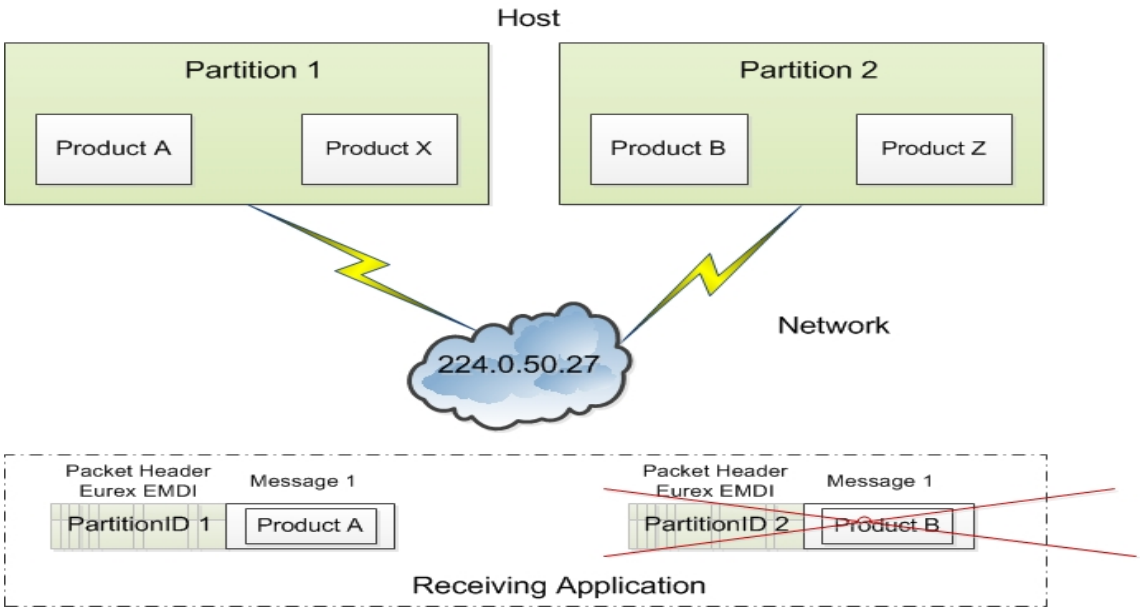

Figure 16: *Discarding packets with unwanted products based on the PartitionID of the T7 EMDI packet header*

The example provided in figure 16 shows two products arriving on multicast address 224.0.50.27. The participant is only interested in product A. Packets containing product A or product X can be identified by the field *PartitionID* in the *packet header*. As product X is not one of the desired products it can be discarded after decoding the message.

Based on the reference data, the receiving application knows that packets coming from PartitionID 2 contain only undesired products. It discards all packets with *PartitionID* = 2 in the *packet header* without decoding message 1.

Part III
Reference

11 Detailed data feed description and layout

This chapter provides message layouts and field information. It is structured by service messages, data messages and data files.

Please consider, that the following tables will only list valid values for enum and set data types, which are used within that specific context. The complete list and order of all valid values supported by a specific enum or set datatype could be found within the T7 Market and Reference Data Interfaces - XML FAST Templates. These files could be found at

www.eurex.com > Support > Initiatives & Releases > T7 Release 13.1 > System Documentation > Market & Reference Data Interfaces

or

www.xetra.com > Technology > T7 trading architecture > System documentation > Release 13.1 > Market and Reference Data Interfaces.

Specifically the actual wire values for Fast 1.1 decoders need to be derived from the XML Fast Templates.

11.1 Service messages

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

11.1.1 FAST reset message

The template with ID = 120 is not included in the “FAST Message Templates” file. This TID is reserved in the main FAST specification and allocated by the FAST Session Control Protocol specification (SCP 1.1[21])

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

11.1.2 Packet header (T7 EMDI)

Delivered in: Every UDP-datagram

The *packet header* is a technical header used for identification of datagrams and is sent on a channel basis. Every partition stamps outgoing datagrams with a sequence number (field: *PacketSeqNum*).

One method to identify duplicates between Service A and B is by the use of the field *PacketSeqNum* which is unique per *senderCompID*; a faster way is to perform a memory comparison on the first 10 bytes of the *Packet header*.

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

[21] FAST Session Control Protocol.

As the *Packet header* message is not defined in the FIX standard, the FIX Tags for *PacketSeqNumber*, *SendingTime* and *PerformanceIndicator* are not shown in the table below. The following layout is available after FAST decoding of the *packet header*:

Field Name	Data Type	Description
PartitionID	uInt32	Sending partition.
SenderCompID	uInt32	Unique id for a sender.
PacketSeqNumber	byte vector	Datagram sequence number.
SendingTime	byte vector	Time when market data feed handler writes packet on the wire.
PerformanceIndicator	byte vector	Current load of system. Time difference between the incoming ETI-order/quote and the time the market data is written to the socket. This information is provided for the incremental feed of T7 EMDI only and is not provided for the T7 MDI. This field should be interpreted as a signed 32-bit integer having a minimum value of 0x80000000 (in case of time synchronisation anomalies the value can be negative).

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

1 Byte	1 Byte	1 Byte	2 Bytes	1 Byte	4 Bytes	1 Byte	8 Bytes	1 Byte	4 Bytes
PMAP	**TID**	**PartitionID**	**SenderCompID** [128 ... 16383]	Length	**PacketSeqNum**	Length	**SendingTime**	Length	**PerformanceIndicator**
1	2	3	4	6	10	11	19	20	24

Figure 17: *Structure of the packet header for T7 EMDI*

The last three fields are byte vectors with fixed length. Byte vectors are not stop bit encoded according to the FAST standard. Each of them are preceded by a FAST encoded 1 Byte length field as per the FAST specification for byte vector fields.

Note: The field *PerformanceIndicator* including the length field is only available in messages on the T7 EMDI incremental feed. The *PartitionID* is available in messages on both incremental and snapshot feed of the T7 EMDI.

11.1.3 Packet header (T7 MDI /T7 RDI)

Delivered in: every UDP-datagram

The *packet header* of T7 MDI and T7 RDI doesn't contain the *PerformanceIndicator* with length field and the *PartitionID*. The rest of the *packet header* is identical to T7 EMDI. Duplicates between Service A and Service B can be detected by a memory comparison on the first 8 bytes of the *packet header*.

Field Name	Data Type	Description
SenderCompID	uInt32	Unique id for a sender
PacketSeqNumber	byte vector	Datagram sequence number
SendingTime	byte vector	Time when market data feed handler writes packet on the wire.

Wire representation:

1 Byte	1 Byte	1 Byte	1 Byte	4 Bytes	1 Byte	8 Bytes
PMAP	**TID**	**SenderCompID** [21 ... 127]	Length	**PacketSeqNum**	Length	**SendingTime**
1	2	3	4	8	9	17

Figure 18: *Structure of the packet header for T7 MDI and T7 RDI*

11.1.4 Functional beacon message

Delivered on: T7 EMDI incremental and T7 RDI incremental

The *functional Beacon* message is sent as a "line active" indicator whenever there are no messages generated on the EMDI incremental feed for the respective product within the last 10 seconds in production and only if the respective product is in *MarketSegmentStatus (2542)* 1 = Active (see 11.2.1, pg. 86). On the RDI incremental feed it is sent every two minutes whenever there are no messages generated.

Functional beacons are sent once the market data service becomes available. If no messages have been sent on the incremental feed for a product (or market for RDI) then *LastMsgSeqNumProcessed (369)* is set to zero.

US-customers receive a *functional beacon* on the EMDI incremental for US-tradable products only.

Tag	Field Name	Req'd	Data Type	Description
35	MsgType	Y	string	**Value** / **Description**: 0 – Beacon
49	SenderCompID	Y	uInt32	Unique id of a sender.
50	SenderSubID	Y	uInt32	Product Identifier, e.g. 89, for EMDI or Market Identifier, e.g. 1 (EUREX), for RDI.
369	LastMsgSeqNum-Processed	Y	uInt32	Last sequence number on the incremental feed for this SenderSubID.

11.1.5 Technical heartbeat message

Delivered on: every channel for T7 EMDI, T7 MDI and T7 RDI

The *technical heartbeat or beacon messages* are transmitted periodically on every multicast address on a specific port assigned for the *technical heartbeat*. The technical heartbeat messages have a variable length payload, which always begins with two magic bytes, 0xC0 0xF8 (FAST Reset). However, the content of the technical heartbeat messages is irrelevant and should be ignored. The messages are sent with the primary purpose to maintain the routing trees, which prevents routers from dropping multicast packages.

During weekends, technical heartbeat messages are sent with a significantly larger payload and higher frequency. This allows participants to detect CRC errors on network connections, for example, after infrastructure maintenance.

11.1.6 Market data report message

Delivered on: T7 RDI snapshot feed

The number of reference data could be large, and it might take some time to complete one snapshot cycle. In order to mark the beginning of the reference data messages, the market data report message is sent with a start- and end-flag indicated by the field *MDReportEvent (2535)*. The fields *MDReportCount (2536)* and *LastMsgSeqNumProcessed (369)* allow to determine the number of snapshots and incrementals within the reference data snapshot feed. For further details see section 9.2.2, Counters as part of the market data report message, on page 46.

Tag	Field Name	Req'd	Data Type	Description
35	MsgType	Y	string	**Value** / **Description**: DR – Market Data Report
2536	MDReportCount	N	uInt32	Number of reference data snapshot messages in the snapshot cycle of the current business day. Only sent for MDReportEvent = 1, i.e. at the start of the cycle.
50	SenderSubID	Y	uInt32	Market Identifier, e.g. 1 for "XEUR".
1301	MarketID	Y	string	Market Identifier Code as specified in ISO 10383.
369	LastMsgSeqNumProcessed	N	uInt32	*MDReportCount* value of the last reference data message (snapshot or incremental) at the end of the snapshot cycle (products and instruments share a single sequence). Number of incremental updates in snapshot cycle can be calculated as *LastMsgSeqNumProcessed* - *MDReportCount*. Only sent for MDReportEvent = 1, i.e. at the start of the cycle.
2535	MDReportEvent	Y	MDReportEvent (enum)	Defines start/end of reference data. Other values have been added due to exchange wide FAST template alignments. **Value** / **Description**: 1 – Start of reference data; 2 – End of reference data
60	TransactTime	Y	timestamp	Creation time of the snapshot.
2537	TotNoMarketSegmentReports	N	uInt32	Contains the number of product level messages sent in the snapshot. Only sent for MDReportEvent = 1.
2538	TotNoInstrumentReports	N	uInt32	Contains the number of instrument level messages sent in the snapshot. Only sent for MDReportEvent = 1.

11.2 Reference data messages

Reference data is delivered via the snapshot and incremental feeds with the T7 RDI. A *product incremental* message does not exist in this release.

11.2.1 Product snapshot message

Delivered on: T7 RDI snapshot feed

The *Product snapshot* message sends the collapsed view of the market, market segment and product. It provides the dynamic multicast addresses/ports for each type (field: *MDFeedType* (1022)) of the market data feed.

The maximum order book depth is specified by the field *MarketDepth* (264).

One message per product is sent.

Tag	Field Name	Req'd	Data Type	Description
35	MsgType	Y	string	Reference data snapshot (one per product). **Value** / **Description**: BU – Market Definition
34	MsgSeqNum	Y	uInt32	Contiguous across all messages on the reference data snapshot feed regardless of message type. See figure 9 chapter 9.2.1, General structure of the snapshot cycle.
1301	MarketID	Y	string	Market Identifier Code as specified in ISO 10383.
1300	MarketSegmentID	Y	uInt32	Product identifier, e.g. "89". Can also be a product pool id.
2400	EffectiveBusinessDate	Y	uInt32	Current business date, e.g. "20131128".
28871	NextEffectiveBusiness-Date	Y	uInt32	Next business date.
7703	MarketSegment	Y	string	Product name, e.g. "FDAX".
2542	MarketSegmentStatus	Y	Market-Segment-Status (enum)	Defines if on-exchange trading on T7 is available for a product (1 = Active) or not (3 = Published). Reason for a published product can be for the following reasons: • The product is still traded on the current platform prior to migration, • First trading day of the new product has not yet been reached, or • the product is setup on the current T7 platform for Off-Book trading with no intention for on-exchange trading. **Value** / **Description**: 1 – Active; 3 – Published **Please note** Market Data specifically functional beacons are only sent on the specified addresses for 1 = Active.
5948	PartitionID	Y	uInt32	Partition of the product.
25246	CapacityGroupID	Y	uInt32	Capacity Group Identifier of the product.
31629	NegotiationDuration	N	uInt32	Contains the default and maximum duration of negotiation event in seconds.
<DerivativesDescriptor> (optional) group starts				
15	Currency	Y	string	Currency as published in ISO 4217. Please note, "GBX" may be used to denote Penny sterling.
1396	MarketSegmentDesc	N	string	Product Description, e.g. "OPT ON THE DAX INDEX".
7177	MarketSegmentSymbol	N	string	Product ISIN, e.g. "DE0009653147".

Tag	Field Name	Req'd	Data Type	Description
1325	ParentMktSegmID	N	string	Standard Eurex product types, e.g. "FINX". For Product Pools the following values will be used: **"FIPS"** for Inter Product Spreads (IPS). **"BTRF"** for BTRF Buckets. **"EBB"** for EBB Buckets.
39543	USApproval	N	USApproval (enum)	Flag to identify whether product may be traded from the USA. **Value** / **Description**: None – No Approval; CFTC – CFTC Approval; SEC – SEC Approval
2543	MarketSegmentType	N	Market-Segment-Type (enum)	**Value** / **Description**: 1 – Pool
2544	MarketSegmentSubType	N	Market-Segment-Sub-Type (enum)	**Value** / **Description**: 1 – Inter Product Spread; 2 – BTRF Bucket; 3 – EBB Bucket; 4 – Commodity Group
25144	DecaySplit	N	DecaySplit (enum)	**Value** / **Description**: 3 – Quarter; 6 – Season; 12 – Year
30126	MaxOffsetRFQExpire-Time	N	uInt32	Defines the max expiry time in seconds describing when the RFQ session will be automatically terminated. RFQ expiry time based on exchange defined *NegotiationDuration (31629)* or by the requester shall be less than this. This attribute is computed as StartOfDayTime + MaxOffsetRFQExpiryTime.
30168	OffsetSTPEffectiveTime	N	uInt32	Defines the waiting period of the final STP deal inside Eurex EnLight in seconds.
28798	TslMarketGroupID	N	uInt32	ID of the Transaction Size Limit (TSL) product group this product belongs to.
28799	TslMarketGroup	N	string	Symbol of the Transaction Size Limit (TSL) product group.
28903	IndependentSystemOp-erator	N	string	Name of the Independent System Operator (ISO) such as PJM, ERCOT, ISONE, NYISO, CAISO, etc. . .
283	LocationID	N	string	Location/ nodes within the ISO which this product covers such as Western Hub, Jersey Central Power, etc . . .
6958	ProductType	N	string	A textual description of the Commodity Type.
28902	CommodityProductClass	N	string	Information about the hours covered by the product such as Peak, Off-Peak, 2x16, 7x8
30996	CommodityUnit	N	uInt32	Information about the load such as 5 MW, 25 MW, 50 MW
30283	LocationType	N	string	Information about the type of Location such as Hub, Zone, Generator node, Interface, Aggregate, Scheduling Point, DefaultLAP, Price Point, Resource Node, Sublap etc.
1140	MaxTradeVol	N	decimal	Maximum Order Quantitiy.
28804	MaxTradeVal	N	uInt64	Maximum Order Value.

Tag	Field Name	Req'd	Data Type	Description
28805	MaxCalendarSpreadVol	N	decimal	Maximum Calendar Spread Quantitiy.
28806	MaxTESVol	N	decimal	Maximum TES Quantity.
<QuantityScalingFactors> (optional) sequence starts				
28906	NoQuantityScalingFactors	N	length	Number of following valid quantity scaling factors for scaled simple instruments.
28907	> QuantityScalingFactor	Y	uInt32	A valid quantity scaling factor.
<QuantityScalingFactors> (optional) sequence ends				
<DerivativesDescriptor> (optional) group ends				
<UnderlyingDescriptor> (optional) group starts				
30308	UnderlyingSecurityExchange	N	string	Market Identifier Code of the underlying as specified in ISO 10383.
30311	UnderlyingSymbol	N	string	Underlying security symbol, e.g. "DAX".
30309	UnderlyingSecurityID	N	string	ISIN code of the underlying, e.g. "DE0008469008".
30305	UnderlyingSecurityIDSource	N	string	**Value** / **Description**: 4 – ISIN
28843	UnderlyingPrevClosePx	N	decimal	Closing price of the underlying on the previous day.
<UnderlyingDescriptor> (optional) group ends				
<InstrumentScopes> sequence starts				
1656	NoInstrumentScopes	N	length	Number of instrument scope in the segment. Always 1.
1535	> InstrumentScopeOperator	Y	InstrumentScopeOperator (enum)	Instrument scope operator. **Value** / **Description**: 1 – Include
1547	> InstrumentScopeSecurityType	Y	InstrumentScopeSecurityType (enum)	Type of security. **Value** / **Description**: MLEG – Complex Instrument; VAR – Variance Futures; TRF – Total Return Futures; TARP – Trade At Reference Price
1548	> InstrumentScopeSecuritySubType	N	uInt32	Standard strategy type for complex instruments. The mapping of integer values to the corresponding strategy types is available at www.eurex.com > Data > Trading files > Product parameter files > Trading parameters
<InstrumentScopes> sequence ends				
<RelatedMarketSegments> sequence starts				
2545	NoRelatedMarketSegments	N	length	Number of market segments having a relationship with the current entry.
2546	> RelatedMarketSegmentID	Y	uInt32	Market segment identifier.

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>2547</td><td>> MarketSegmentsRelationship</td><td>Y</td><td>Market-Segment-Relationship (enum)</td><td>Defines the type of the relationship.
102 = BTRF Bucket Product of a BTRF Bucket.
103 = EBB Bucket Product of a Equity Bespoke Bucket.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>100</td><td>Cash leg for volatility strategies.</td></tr><tr><td>101</td><td>Target Product for Decaying Product.</td></tr><tr><td>102</td><td>BTRF Bucket</td></tr><tr><td>103</td><td>EBB Bucket</td></tr><tr><td>104</td><td>Commodity Group</td></tr></table></td></tr>
<tr><td>28901</td><td>> AllowOneProductStrip</td><td>N</td><td>Allow-One-Product-Strip (enum)</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>One Product Strip not allowed.</td></tr><tr><td>1</td><td>One Product Strip allowed.</td></tr></table></td></tr>
<tr><td colspan="5"><RelatedMarketSegments> sequence ends</td></tr>
<tr><td colspan="5"><BaseTrdgRules> group starts</td></tr>
<tr><td colspan="5"><TickRules> sequence starts</td></tr>
<tr><td>1205</td><td>NoTickRules</td><td>Y</td><td>length</td><td>Number of tick rules defining the price step table for this product.
See T7 Functional Reference v 7.1.0 (or greater), paragraph about “Limit Price”.</td></tr>
<tr><td>28887</td><td>> TickRuleID</td><td>Y</td><td>uInt32</td><td>Price Step Table identifier. This is used to identify all tick rules that belong to the same price step table alias tick rule table. A price step table may consist of one or more tick rules with the same TickRuleID (28887).
For cash market products for on book trading it is also used to reference the applicable tick rule table by RefTickTableID (1787) from any Instrument Snapshot message.</td></tr>
<tr><td>1206</td><td>> StartTickPriceRange</td><td>Y</td><td>decimal</td><td>Starting price range for specified tick increment (inclusive).</td></tr>
<tr><td>1207</td><td>> EndTickPriceRange</td><td>Y</td><td>decimal</td><td>Ending price range for the specified tick increment (non-inclusive).</td></tr>
<tr><td>1208</td><td>> TickIncrement</td><td>Y</td><td>decimal</td><td>Tick increment for stated price range. Specifies the valid price increments at which quotes or trades for an instrument of the product can be entered.</td></tr>
<tr><td colspan="5"><TickRuleScopes> sequence starts</td></tr>
<tr><td>32571</td><td>> NoTickRuleScopes</td><td>Y</td><td>length</td><td>Number of tick rule scope definitions.
Each combination of TickRuleProductComplex (2571) and TrdType (828) defines a valid scope for which this tick rule is used for.
Please note, that for cash market products a tick rule with a TickRuleID (28887) may additionally be referenced by the same RefTickTableID (1787) for on book trading as well (see Instrument snapshot message 96.)</td></tr>
</table>

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>2571</td><td>> > TickRuleProductComplex</td><td>Y</td><td>Product-Complex (enum)</td><td>Defines the instrument type for this tick rule scope.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Simple Instrument</td></tr><tr><td>2</td><td>Standard Option Strategy</td></tr><tr><td>3</td><td>Non-Standard Option Strategy</td></tr><tr><td>4</td><td>Volatility Strategy</td></tr><tr><td>5</td><td>Futures Spread</td></tr><tr><td>6</td><td>Inter-Product Spread</td></tr><tr><td>7</td><td>Standard Futures Strategy</td></tr><tr><td>8</td><td>Pack and Bundle</td></tr><tr><td>9</td><td>Strip</td></tr><tr><td>10</td><td>Flexible Instrument</td></tr><tr><td>11</td><td>Commodity Strips</td></tr><tr><td>12</td><td>Scaled Simple Instrument</td></tr><tr><td>13</td><td>Non-Standard Volatility Strategy</td></tr><tr><td>14</td><td>Total Return Future Strategy</td></tr></table></td></tr>
<tr><td>828</td><td>> > TrdType</td><td>Y</td><td>TrdType (enum)</td><td>Defines the (TES) trade type for this tick rule scope.
0 = Regular Trade is used for derivatives market products to refer to the tick rule table used for regular on book trading and is also used as the default for TES.
Only TES trade types with a deviating tick rule table from the on book tick rule table of the corresponding TickRuleProduct-Complex (2571) are given explicitly in TrdType (828).
1006 = Xetra / Eurex Enlight triggered Trade Indicates a Eurex Enlight triggered trade.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Regular Trade</td></tr><tr><td>1</td><td>Block Trade / Large in Scale (LIS)</td></tr><tr><td>12</td><td>EFS</td></tr><tr><td>54</td><td>OTC</td></tr><tr><td>1000</td><td>Vola</td></tr><tr><td>1001</td><td>EFP-Fin</td></tr><tr><td>1002</td><td>EFP-Index</td></tr><tr><td>1004</td><td>Block Tam</td></tr><tr><td>1005</td><td>LIS</td></tr><tr><td>1006</td><td>Xetra / Eurex Enlight triggered Trade</td></tr><tr><td>1007</td><td>Block QTPIP Trade</td></tr><tr><td>1017</td><td>Delta Trade At Market</td></tr></table></td></tr>
<tr><td colspan="5"><TickRuleScopes> sequence ends</td></tr>
<tr><td colspan="5"><TickRules> sequence ends</td></tr>
<tr><td colspan="5"><PriceRangeRules> sequence starts</td></tr>
<tr><td>2550</td><td>NoPriceRangeRules</td><td>Y</td><td>length</td><td>Price Range Tables.
See T7 Functional Reference, section “Price Range Table Basics” available at
www.eurex.com > Support > Initiatives & Releases > T7 Release 13.1 > System Documentation > Overview and Functionality</td></tr>
<tr><td>2556</td><td>> PriceRangeRuleID</td><td>Y</td><td>uInt32</td><td>Table identifier.</td></tr>
</table>

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>2555</td><td>> PriceRangeProduct-Complex</td><td>Y</td><td>Product-Complex (enum)</td><td>Not filled for cash.<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Simple Instrument</td></tr><tr><td>2</td><td>Standard Option Strategy</td></tr><tr><td>3</td><td>Non-Standard Option Strategy</td></tr><tr><td>4</td><td>Volatility Strategy</td></tr><tr><td>5</td><td>Futures Spread</td></tr><tr><td>6</td><td>Inter-Product Spread</td></tr><tr><td>7</td><td>Standard Futures Strategy</td></tr><tr><td>8</td><td>Pack and Bundle</td></tr><tr><td>9</td><td>Strip</td></tr><tr><td>10</td><td>Flexible Instrument</td></tr><tr><td>11</td><td>Commodity Strips</td></tr><tr><td>12</td><td>Scaled Simple Instrument</td></tr><tr><td>13</td><td>Non-Standard Volatility Strategy</td></tr><tr><td>14</td><td>Total Return Future Strategy</td></tr></table></td></tr>
<tr><td>2551</td><td>> StartPriceRange</td><td>Y</td><td>decimal</td><td>Start of price range (inclusive).</td></tr>
<tr><td>2552</td><td>> EndPriceRange</td><td>Y</td><td>decimal</td><td>End of price range (non-inclusive).</td></tr>
<tr><td>2553</td><td>> PriceRangeValue</td><td>N</td><td>decimal</td><td>Maximum allowable quote spread (absolute value). Conditionally required if PriceRangePercentage is absent.</td></tr>
<tr><td>2554</td><td>> PriceRangePercentage</td><td>N</td><td>decimal</td><td>Maximum allowable quote spread (percentage value). Conditionally required if PriceRangeValue is absent.</td></tr>
<tr><td colspan="5"><PriceRangeRules> sequence ends</td></tr>
<tr><td>2559</td><td>QuoteSideIndicator</td><td>N</td><td>QuoteSide-Indicator (enum)</td><td>Defines if single sided quotes are allowed on entry.
Only applicable for MarketSegmentSubType (2544) 1 = BTRF Bucket and 3 = EBB Bucket.<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>One-sided quote not allowed</td></tr><tr><td>1</td><td>One-sided quote allowed</td></tr></table></td></tr>
<tr><td>28898</td><td>QuoteSideModelType</td><td>N</td><td>QuoteSide-ModelType (enum)</td><td>Defines if a single sided quote rejection leads to a double sided quote rejection or cancellation.
0 = Single-sided quotes not supported Rejection of just one quote side leads to rejection or cancellation of both quote sides.
1 = Single-sided quotes supported Rejection may affect only one quote side in a double sided quote.
Not applicable for MarketSegmentSubType (2544) 2 = BTRF Bucket and 3 = EBB Bucket.<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Single-sided quotes not supported</td></tr><tr><td>1</td><td>Single-sided quotes supported</td></tr></table></td></tr>
</table>

Tag	Field Name	Req'd	Data Type	Description
2557	FastMarketPercentage	N	decimal	Percentage by which range resulting from *PriceRangeValue (2553)* and *PriceRangePercentage (2554)* has to be extended to obtain the valid price range during Fast Market, or Stressed Market Conditions in derivatives markets: 0 = no change to normal range, e.g. 80->80 50 = increase by half, e.g. 80->120 100 = range to be doubled, e.g. 80->160 More details are available in the T7 Functional Reference available at www.eurex.com > Support > Initiatives & Releases > T7 Release 13.1 > System Documentation > Overview and Functionality
25217	CheckMarketOrder	N	Check-Market-Order (enum)	Specifies if Market Orders will be checked against the available bid/ask price on the opposing side in specifc scenarios. Only used for derivatives markets products. **Value** / **Description** N / Do Not Check Y / Check
<HHIIntervals> (optional) sequence starts				
25227	NoHHIIntervals	Y	length	Herfindahl-Hirschman Index (HHI) intervals.
25228	> HHIIndicator	Y	uInt32	The HHIInterval value.
25229	> HHIIntervalEnd	Y	decimal	The HHIInterval upper boundary (incl.), e.g. HHIIndicator $= 1$ for $[0, \text{HHIIntervalEnd}_1]$, HHIIndicator $= n$ for $(\text{HHIIntervalEnd}_{(n-1)}, \text{HHIIntervalEnd}_n]$, for $n > 1$.
<HHIIntervals> (optional) sequence ends				
<BaseTrdgRules> group ends				
<BaseTrdgRulesDerivatives> (optional) group starts				
<QuoteSizeRules> (optional) sequence starts				
2558	NoQuoteSizeRules	N	length	
647	> MinBidSize	N	decimal	Minimum bid quantity (identical to minimum offer quantity).
648	> MinOfferSize	N	decimal	Minimum offer quantity (identical to minimum bid quantity).
2447	> FastMarketIndicator	N	FastMarket-Indicator (enum)	Indicates if the values are specific to Fast Market, or to Stressed Market Conditions in derivatives markets. **Value** / **Description** 0 / No 1 / Yes
<QuoteSizeRules> (optional) sequence ends				
<FlexRules> sequence starts				
2560	NoFlexProductEligibilities	N	length	

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>2561</td><td>> FlexProductEligibility-Complex</td><td>Y</td><td>Product-Complex (enum)</td><td>Defines the strategy types available for user defined strategies.<table><tr><th>Value</th><th>Description</th></tr><tr><td>2</td><td>Standard Option Strategy</td></tr><tr><td>3</td><td>Non-Standard Option Strategy</td></tr><tr><td>4</td><td>Volatility Strategy</td></tr><tr><td>5</td><td>Futures Spread</td></tr><tr><td>6</td><td>Inter-Product Spread</td></tr><tr><td>7</td><td>Standard Futures Strategy</td></tr><tr><td>8</td><td>Pack and Bundle</td></tr><tr><td>9</td><td>Strip</td></tr><tr><td>10</td><td>Flexible Instrument</td></tr><tr><td>13</td><td>Non-Standard Volatility Strategy</td></tr></table></td></tr>
<tr><td>1242</td><td>> FlexProductEligibilityIndicator</td><td>Y</td><td>Flex-Product-Eligibility-Indicator (enum)</td><td>Defines if user defined strategies are allowed.<table><tr><th>Value</th><th>Description</th></tr><tr><td>N</td><td>No</td></tr><tr><td>Y</td><td>Yes</td></tr></table></td></tr>
<tr><td colspan="5"><FlexRules> sequence ends</td></tr>
<tr><td>561</td><td>RoundLot</td><td>N</td><td>decimal</td><td>Minimum Tradable Unit in derivatives markets.</td></tr>
<tr><td colspan="5"><BaseTrdgRulesDerivatives> (optional) group ends</td></tr>
<tr><td colspan="5"><BaseTrdgRulesCash> (optional) group starts</td></tr>
<tr><td colspan="5"><AuctionTypeRuleGrp> sequence starts</td></tr>
<tr><td>2548</td><td>NoAuctionTypeRules</td><td>Y</td><td>length</td><td>Number of rules. Always 1.</td></tr>
<tr><td>1803</td><td>> AuctionType</td><td>Y</td><td>AuctionType (enum)</td><td>Defines the type of auction.<table><tr><th>Value</th><th>Description</th></tr><tr><td>100</td><td>Any Auction</td></tr></table></td></tr>
<tr><td>28874</td><td>> ClosedBookIndicator</td><td>Y</td><td>Closed-Book-Indicator (enum)</td><td>Indicates whether the order book is closed during auction trading.
N = No applicable for cash market products only.<table><tr><th>Value</th><th>Description</th></tr><tr><td>N</td><td>No</td></tr><tr><td>Y</td><td>Yes</td></tr></table></td></tr>
<tr><td>28875</td><td>> MarketImbalanceIndicator</td><td>Y</td><td>Market-Imbalance-Indicator (enum)</td><td>Controls if during auction call/volatility interruption/extended volatility interruption/market order interruption phase a surplus (side and volume) at the indicative price (if crossed order book) or the best bid/best ask limit and quantity (if uncrossed order book) is displayed to the market. Only makes sense when ClosedBookIndicator = Y.
Y = Yes applicable for cash market products only.<table><tr><th>Value</th><th>Description</th></tr><tr><td>N</td><td>No</td></tr><tr><td>Y</td><td>Yes</td></tr></table></td></tr>
<tr><td colspan="5"><AuctionTypeRuleGrp> sequence ends</td></tr>
</table>

Tag	Field Name	Req'd	Data Type	Description
28800	DisableOnBookTrading	Y	DisableOnBookTradingFlag (enum)	Defines, if On Book trading is disabled. **Value** / **Description** N – No Y – Yes
<BaseTrdgRulesCash> (optional) group ends				
<MatchRules> sequence starts				
1235	NoMatchRules	Y	length	Used to convey allocation rules for matching
2569	> MatchRuleProductComplex	Y	ProductComplex (enum)	Indicates the instrument type. **Value** / **Description** 1 – Simple Instrument 2 – Standard Option Strategy 3 – Non-Standard Option Strategy 4 – Volatility Strategy 5 – Futures Spread 6 – Inter-Product Spread 7 – Standard Futures Strategy 8 – Pack and Bundle 9 – Strip 11 – Commodity Strips 12 – Scaled Simple Instrument 13 – Non-Standard Volatility Strategy
1142	> MatchAlgorithm	Y	MatchAlgorithm (enum)	Defines the order allocation method. **Value** / **Description** PT – Price-time PR – Pro-rata TPR – Time pro-rata
574	> MatchType	N	MatchType (enum)	If *MatchType* is not filled the same matching algorithm is used for continuous trading and auction trades. AutoMatch stands for continuous trading and CallAuction for auction trades. **Value** / **Description** 4 – Auto Match 7 – Call Auction
<MatchRules> sequence ends				
<Feeds> sequence starts				
1141	NoMDFeedTypes	Y	length	Number of feeds.
1022	> MDFeedType	Y	MDFeedType (enum)	Type of feed. **Value** / **Description** HI – High Incremental (unnetted feed) HS – High Snapshot (unnetted feed) L – Low (netted feed)
1021	> MDBookType	Y	MDBookType (enum)	Type of book data. **Value** / **Description** 2 – Price Depth 3 – Order Depth

Tag	Field Name	Req'd	Data Type	Description
264	> MarketDepth	N	uInt32	Order book (price level) depth.
2563	> MarketDepthTimeInterval	N	uInt32	Netting interval in milliseconds for low bandwidth feeds (0=no netting).
2565	> MDRecoveryTimeInterval	N	uInt32	Recovery interval in milliseconds (duration of one cycle).
2567	> PrimaryServiceLocationID	Y	string	IP Address for Service A.
28591	> PrimaryServiceLocationSubID	Y	uInt32	Port number for IP address Service A.
2568	> SecondaryServiceLocationID	N	string	IP Address Service B.
28593	> SecondaryServiceLocationSubID	N	uInt32	Port number for IP address Service B.
<Feeds> sequence ends				

11.2.2 Instrument snapshot message

Delivered on: T7 RDI snapshot feed

The *Instrument snapshot* message is used to describe simple instruments.

A link to the reference information of the T7 System is provided with the field *SecurityAltID (455)*.
One message per instrument is sent.

Tag	Field Name	Req'd	Data Type	Description
35	MsgType	Y	string	**Value** / **Description**: d – Security Definition
34	MsgSeqNum	Y	uInt32	Contiguous across all messages on the reference data snapshot feed regardless of message type. See figure 9 chapter 9.2.1, General structure of the snapshot cycle.
48	SecurityID	Y	int64	The instrument identifier uniquely identifies an instrument, e.g. 72057800196358145.
22	SecurityIDSource	Y	string	**Value** / **Description**: M – Marketplace-assigned identifier
<SecurityAlt> (optional) sequence starts				
454	NoSecurityAltID	Y	length	Number of alternate identifiers.
455	> SecurityAltID	Y	string	Alternate instrument identifier with respect to *SecurityAltIDSource (456)*.
456	> SecurityAltIDSource	Y	string	Identifies class or source of the SecurityAltID (455) value. **4 = ISIN** Always present for simple instruments. **8 = Exchange Symbol** set to the instruments *mnemonic*, only applicable for cash market instruments. **B = Wertpapier (WKN = Wertpapierkennnummer)** only applicable for cash market instruments. **M = Marketplace-assigned identifier** only applicable for derivatives market instruments (unique identifier across T7, C7, Prisma, aka ContractID). **102 = Redundant To SecurityID** The *SecurityAltID(455)* will point to the instrument with the same *SecurityID(48)* which "replaces" this inactivated instrument. **Value** / **Description**: 4 – ISIN; 8 – Exchange Symbol; B – Wertpapier (WKN = Wertpapierkennnummer); M – Marketplace-assigned identifier; 102 – Redundant To SecurityID
<SecurityAlt> (optional) sequence ends				

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>167</td><td>SecurityType</td><td>Y</td><td>Security-Type (enum)</td><td>Type of security.
SR = Subscription Right only applicable for cash market instruments.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>OPT</td><td>Option</td></tr>
<tr><td>FUT</td><td>Future</td></tr>
<tr><td>MLEG</td><td>Complex instrument</td></tr>
<tr><td>VAR</td><td>Variance Future</td></tr>
<tr><td>TRF</td><td>Total Return Future</td></tr>
<tr><td>CS</td><td>Common stock</td></tr>
<tr><td>BOND</td><td>Bond</td></tr>
<tr><td>TARP</td><td>Trade at reference price</td></tr>
<tr><td>ETF</td><td>Exchange Traded Fund</td></tr>
<tr><td>ETC</td><td>Exchange Traded Commodity</td></tr>
<tr><td>ETN</td><td>Exchange Traded Note</td></tr>
<tr><td>WAR</td><td>Warrant</td></tr>
<tr><td>OTHER</td><td>Other</td></tr>
<tr><td>SR</td><td>Subscription Right</td></tr>
<tr><td>FUN</td><td>Investment Funds</td></tr>
</table></td></tr>
<tr><td>965</td><td>SecurityStatus</td><td>Y</td><td>Security-Status (enum)</td><td>Instruments may be published prior to becoming active for trading. Status is identical to the product level, i.e. all instruments of a product are either published or active with the exception of those that expire one or more business days after they stop trading. These will still be distributed until they expire but as inactive.
Active instruments can also expire intraday or be suspended which is conveyed with the same field but only within intraday State Change messages (EMDI or MDI).
2 = Inactive is used for cash instruments when the first trading date is in future and the product state is Active. The SecurityStatus switches to Active as soon the first trading date is equal or later than current business date.
11 = Pending Deletion is used for cash instruments when the last trading date is exceeded and the instrument will be deleted soon. Those instruments might become 1 = Active again intraday. This change will only be conveyed with Market Data intraday State Change messages and won't affect the Instrument Snapshot on the Reference Data Interface.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>Active</td></tr>
<tr><td>2</td><td>Inactive</td></tr>
<tr><td>10</td><td>Published</td></tr>
<tr><td>11</td><td>Pending Deletion</td></tr>
</table></td></tr>
</table>

Tag	Field Name	Req'd	Data Type	Description
107	SecurityDesc	N	string	For derivatives market products the *SecurityDesc (107)* is a unique business identifier. The format depends on the *ProductComplex (1227)*: Simple Instrument: *PROD* SI *YYYYMMDD SM ES* C/P *StrikePrice VN* Standard options strategies: *PROD*.O.*YYMMDD.IST.SEQ-NO* Non-standard options strategies: *PROD*.N.*YYMMDD.SEQ-NO* Options volatility strategies: *PROD*.V.*YYMMDD.IST.SEQ-NO* Future calendar spread: *PROD*.S.*MONYY.MONYY*.SPD Basis calendar spread: *PROD*.S.*YYMMDD.MONYY*.BSPD Inter product spreads: *PROD*.I.*YYMMDD.IST.SEQ-NO* Standard future strategies: *PROD*.F.*YYMMDD.IST.SEQ-NO* Packs & Bundles: *PROD*.B.*MONYY.IST* Strips: *PROD*.T.*YYMMDD.SEQ-NO* Commodity strips: *PROD*.C.*YYMMDD.IST.SEQ-NO* Flexible options: *PROD* FI *YYYYMMDD SM ES* C/P *StrikePrice VN* Flexible futures: *PROD* FI *YYYYMMDD SM* Scaled Simple Instrument: *PROD* SI *YYYYMMDD SM ES* C/P *StrikePrice VN QSF* Non-Standard volatility strategies: *PROD*.D.*YYYYMMDD.SEQ-NO* Total Return Future Strategy: *PROD*.R.*YYYYMMDD.IST.SEQ-NO* where *SM* is the *SettlMethod (1193)* (CS = Cash, PS = Physical), *ES* is the *ExerciseStyle (1194)* (EU = European, AM = American), *VN* is the version number *OptAttribute (206)*, *IST* is the name of the *SecuritySubType (762)* and *SEQ-NO* is a unique sequence number of 6 characters per product. For SI, FI, SPD and BSPD the date *YYYYMMDD* is the *ContractDate (30866)*. For SPD and BSPD *MONYY* is *ContractMonthYear (32340)*. For strategies O, N, V, D, F, Inter product spreads I, Strips T and Commodity strips C the date *YYMMDD* is the creation date. For scaled simple instruments *QSF* is the *QuantityScalingFactor (28907)*. For cash market products, the instrument's long name.
207	SecurityExchange	N	string	MIC (ISO 10383). Used, for derivatives, to identify an instrument of a co-operation partner. For cash this is the exchange where the IPO took place ("Primary Market").

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>1227</td><td>ProductComplex</td><td>Y</td><td>Product-Complex (enum)</td><td>Type of instrument.<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Simple Instrument</td></tr><tr><td>2</td><td>Standard Option Strategy</td></tr><tr><td>3</td><td>Non-Standard Option Strategy</td></tr><tr><td>4</td><td>Volatility Strategy</td></tr><tr><td>5</td><td>Futures Spread</td></tr><tr><td>6</td><td>Inter-Product Spread</td></tr><tr><td>7</td><td>Standard Futures Strategy</td></tr><tr><td>8</td><td>Pack and Bundle</td></tr><tr><td>9</td><td>Strip</td></tr><tr><td>10</td><td>Flexible Instrument</td></tr><tr><td>11</td><td>Commodity Strips</td></tr><tr><td>12</td><td>Scaled Simple Instrument</td></tr><tr><td>13</td><td>Non-Standard Volatility Strategy</td></tr><tr><td>14</td><td>Total Return Future Strategy</td></tr></table></td></tr>
<tr><td colspan="5"><DerivativesDescriptor> (optional) group starts</td></tr>
<tr><td>461</td><td>CFICode</td><td>N</td><td>string</td><td>Indicates the type of security using ISO 10962 standard.</td></tr>
<tr><td>28791</td><td>DisplayName</td><td>N</td><td>string</td><td>See ContractDisplayInstruction (25186), DisplayDay (25210), DisplayRelativeDay (25220), DisplayWeek (25212), DisplayMonth (25211), DisplaySeason (25214) and DisplayYear (25213).</td></tr>
<tr><td>25215</td><td>ContractIdentification-Eligibility</td><td>N</td><td>Contract-Identification-Eligibility (enum)</td><td>Defines the granularity which suffice to identify a standard e.g. non-flexible contract uniquely within a product.<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Contract Month Year</td></tr><tr><td>2</td><td>Expiration Date</td></tr><tr><td>3</td><td>Contract Date</td></tr></table></td></tr>
<tr><td>25216</td><td>IsPrimary</td><td>N</td><td>IsPrimary-Flag (enum)</td><td>Defines, if this contract based on its contract generation cycle, is considered primary, e.g. for XEUR contracts with monthly or higher ContractCycleType (30865).<table><tr><th>Value</th><th>Description</th></tr><tr><td>N</td><td>No</td></tr><tr><td>Y</td><td>Yes</td></tr></table></td></tr>
<tr><td>28907</td><td>QuantityScalingFactor</td><td>N</td><td>uInt32</td><td>Quantity scaling factor, only applicable for scaled instruments (simple or complex).</td></tr>
<tr><td colspan="5"><SimpleInstrumentDescriptor> (optional) group starts</td></tr>
<tr><td>2962</td><td>SecurityReferenceData-Supplement</td><td>N</td><td>uInt32</td><td>Date used to identify the instrument (YYYYMMDD). Same as user defined field ContractDate (30866).</td></tr>
<tr><td>30866</td><td>ContractDate</td><td>N</td><td>uInt32</td><td>Date used to identify the instrument (YYYYMMDD). Same as SecurityReferenceDataSupplement (2962).</td></tr>
<tr><td>32865</td><td>ContractDateType</td><td>N</td><td>string</td><td>A symbol to describe the type of ContractDate (30866) e.g.
EXPI for Expiration Date.
FDLV for First Delivery Date (Power and Gas products of EEX).
FSRD for Final Settlement Reference Date (Eurex MSCI products).
IMMD for IMM Date (start of the underlying interest period) (Eurex ESTR and SARON futures).
HEXP for Home Exchange Expiration Date.</td></tr>
</table>

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>32340</td><td>ContractMonthYear</td><td>N</td><td>uInt32</td><td>Contract month and year of the instrument (YYYYMM).</td></tr>
<tr><td>33865</td><td>ContractMonthType</td><td>N</td><td>Contract-MonthType (enum)</td><td>Provides additional information about ContractMonthYear (32340) and DisplayMonth (25211) (if applicable), e.g.
2 = Maturity Month Plus One The ContractMonthYear (32340) and DisplayMonth (25211) is one month after the month of the MaturityDate (541) or ContractDate (30866).
<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Maturity Month</td></tr><tr><td>2</td><td>Maturity Month Plus One</td></tr><tr><td>3</td><td>IMM Month</td></tr><tr><td>4</td><td>First Delivery Month</td></tr><tr><td>5</td><td>Final Settlement Date Month</td></tr></table></td></tr>
<tr><td>30865</td><td>ContractCycleType</td><td>N</td><td>Contract-CycleType (enum)</td><td>Indicates the kind of regular expiration pattern, in the context of which the instrument has been created by the exchange, e.g. yearly, quarterly, monthly etc. expiration patterns.
Please note, ContractCycleType (30865) might change during the lifetime of an instrument.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Daily</td></tr><tr><td>2</td><td>Weekly</td></tr><tr><td>3</td><td>Monthly</td></tr><tr><td>4</td><td>Quarterly</td></tr><tr><td>5</td><td>SemiAnnually</td></tr><tr><td>6</td><td>Yearly</td></tr></table></td></tr>
<tr><td>31865</td><td>ContractCycleSubType</td><td>N</td><td>Contract-CycleSub-Type (enum)</td><td>Indicates the kind of regular expiration pattern, in the context of ContractCycleType (30865).
1 = EndOfMonth optionally set for ContractCycleType (30865) 2 = Weekly.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>End Of Month</td></tr></table></td></tr>
<tr><td>2982</td><td>MaturityFrequencyUnit</td><td>N</td><td>Maturity-Frequency-Unit (enum)</td><td>Defines how granular the expiration of the contract can take place.
Please note, remains fix during the lifetime of the corresponding instrument. Will always be set to uniquely identify an instrument.
Same as user defined field ContractFrequency (30867).
<table><tr><th>Value</th><th>Description</th></tr><tr><td>D</td><td>Day</td></tr><tr><td>Wk</td><td>Week</td></tr><tr><td>Mo</td><td>Month</td></tr><tr><td>F</td><td>Flexible</td></tr><tr><td>EOM</td><td>End Of Month</td></tr></table></td></tr>
</table>

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>30867</td><td>ContractFrequency</td><td>N</td><td>Contract-Frequency (enum)</td><td>Defines how granular the expiration of the contract can take place.
Please note, ContractFrequency (30867) remains fix during the lifetime of the corresponding instrument. Will always be set to uniquely identify an instrument.
Same as MaturityFrequencyUnit (2982).
<table><tr><th>Value</th><th>Description</th></tr><tr><td>D</td><td>Day</td></tr><tr><td>Wk</td><td>Week</td></tr><tr><td>Mo</td><td>Month</td></tr><tr><td>Flex</td><td>Flex</td></tr><tr><td>EOM</td><td>End Of Month</td></tr></table></td></tr>
</table>

Tag	Field Name	Req'd	Data Type	Description
25186	ContractDisplayInstruction	N	ContractDisplayInstruction (enum)	Contains a recommendation by the exchange, on how to interpret the display attributes *DisplayDay (25210)*, *DisplayRelativeDay (25220)*, *DisplayWeek (25212)*, *DisplayMonth (25211)*, *DisplayQuarter (25189)*, *DisplaySeason (25214)* and *DisplayYear (25213)* and how to construct the name for a simple instrument based on these attributes along with other key instrument attributes. The exchange also provides a recommended *DisplayName (28791)*. The recommended name can be constructed as follows **0 = None** The exchange does not recommend any specific naming for the instrument and provides the DisplayName similar to *SecurityDesc (107)*. Example: `ADS SI 20211217 PS AM C 92.00 0` **1 = Date** The value Date is meant to be used for the Energy Market daily instruments. The exchange recommends a name based on the *ContractDate (30866)* and provides the DisplayName such as <*MarketSegment (7703)*> <*DisplayDayOfWeek (25239)* in DDD format>-<*DisplayDay (25210)*>.<*DisplayMonth (25211)*>.<*DisplayYear (25213)* in YY format> <*StrikePrice (202)*> <*PutOrCall (201)*> <*OptAttribute (206)*>. Example: `ADS FRI-17.12.21 92.00 C` **2 = Month** The value Month is meant to be used for the instruments with monthly expiry. The exchange recommends a name based on *DisplayMonth (25211)* and *DisplayYear (25213)*. The provided DisplayName can be calculated as <*MarketSegment (7703)*> <*DisplayMonth (25211)* in MMM format><*DisplayYear (25213)* in YY format> <*StrikePrice (202)*> <*PutOrCall (201)*> <*OptAttribute (206)*>. Example: `ODAX AUG21 15000 C`, `FDAX DEC21` **3 = Permanent** The exchange recommends specific name for the instrument and provides the DisplayName. The display attributes should not be used. **4 = Quarter** The value Quarter is meant to be used for the Energy Market instruments with quarterly expiry. The exchange recommends a name based on *DisplayQuarter (25189)* and *DisplayYear (25213)*. The provided DisplayName can be calculated as <*MarketSegment (7703)*> Q<*DisplayQuarter (25189)*>-<*DisplayYear (25213)* in YY format> <*StrikePrice (202)*> <*PutOrCall (201)*> <*OptAttribute (206)*>. Example: `H2BQ Q4-21`, `O2BQ Q4-21 72.00 C` **5 = Season** The value Season is meant to be used for the Energy Market instruments with semi-annual expiry. The exchange recommends a name based on *DisplaySeason (25214)* and *DisplayYear (25213)*. The provided DisplayName can be calculated as <*MarketSegment (7703)*> <*DisplaySeason (25214)*>-<*DisplayYear (25213)* in YY format> <*StrikePrice (202)*> <*PutOrCall (201)*> <*OptAttribute (206)*>. Example: `H2BS WIN-21`, `H2BS SUM-22` **6 = Week Of Year** The value Week Of Year is meant to be used for the Energy Market weekly instruments. The exchange recommends a name based on *DisplayWeek (25212)* and *DisplayYear (25213)*. The provided DisplayName can be calculated as <*MarketSegment (7703)*> W<*DisplayWeek (25212)*>-<*DisplayYear (25213)* in YY format> <*StrikePrice (202)*> <*PutOrCall (201)*> <*OptAttribute (206)*>. Example: `DEB1 W05-22`

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td>
7 = Year The value Year is meant to be used for the Energy Market instruments with yearly expiry. The exchange recommends a name based on <i>DisplayYear (25213)</i>. The provided DisplayName can be calculated as <<i>MarketSegment (7703)</i>> <<i>DisplayYear (25213)</i>> <<i>StrikePrice (202)</i>> <<i>PutOrCall (201)</i>> <<i>OptAttribute (206)</i>>.

Example: <code>H2BY 2022</code>, <code>F7PY 2023</code>

8 = Week Of Month The value Week Of Month is meant to be used for weekly instruments. The exchange recommends a name based on <i>DisplayWeek (25212)</i>, <i>DisplayDayOfWeek (25239)</i>, <i>DisplayMonth (25211)</i> and <i>DisplayYear (25213)</i>. The provided DisplayName can be calculated as <<i>MarketSegment (7703)</i>> <<i>DisplayMonth (25211)</i> in MMM format><<i>DisplayYear (25213)</i> in YY format>-W<<i>DisplayWeek (25212)</i>><<i>DisplayDayOfWeek (25239)</i> in DDD format> <<i>StrikePrice (202)</i>> <<i>PutOrCall (201)</i>> <<i>OptAttribute (206)</i>>.

Example: <code>ODAX AUG21-W4FRI 15000 C</code>

9 = Weekend Of Year The value Weekend Of Year is meant to be used for the Energy Market weekly instruments. The exchange recommends a name based on <i>DisplayWeek (25212)</i> and <i>DisplayYear (25213)</i>. The provided DisplayName can be calculated as <<i>MarketSegment (7703)</i>> WE<<i>DisplayWeek (25212)</i>>-<<i>DisplayYear (25213)</i> in YY format> <<i>StrikePrice (202)</i>> <<i>PutOrCall (201)</i>> <<i>OptAttribute (206)</i>>.

Example: <code>DWB1 WE05-22 50.10 C</code>

10 = Relative Day The value Relative Day is meant to be used for daily instruments. The exchange recommends a name based on the number of business days between the <i>ContractDate (30866)</i> and the current business day. The provided DisplayName can be calculated as <<i>MarketSegment (7703)</i>> T±<<i>DisplayRelativeDay (25220)</i>> <<i>StrikePrice (202)</i>> <<i>PutOrCall (201)</i>> <<i>OptAttribute (206)</i>>.

Example: <code>FMWN T+0</code>, <code>FMWN T+1</code>, <code>FMWN T+2</code>

11 = End Of Month The value End Of Month is meant to be used for the instruments with month end expiry. The exchange recommends a name based on <i>DisplayMonth (25211)</i> and <i>DisplayYear (25213)</i>. The provided DisplayName can be calculated as <<i>MarketSegment (7703)</i>> <<i>DisplayMonth (25211)</i> in MMM format><<i>DisplayYear (25213)</i> in YY format>-EOM <<i>StrikePrice (202)</i>> <<i>PutOrCall (201)</i>> <<i>OptAttribute (206)</i>>.

Example: <code>ODAX AUG21-EOM 15000 C</code>

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>None</td></tr>
<tr><td>1</td><td>Date</td></tr>
<tr><td>2</td><td>Month</td></tr>
<tr><td>3</td><td>Permanent</td></tr>
<tr><td>4</td><td>Quarter</td></tr>
<tr><td>5</td><td>Season</td></tr>
<tr><td>6</td><td>Week Of Year</td></tr>
<tr><td>7</td><td>Year</td></tr>
<tr><td>8</td><td>Week Of Month</td></tr>
<tr><td>9</td><td>Weekend Of Year</td></tr>
<tr><td>10</td><td>Relative Day</td></tr>
<tr><td>11</td><td>End Of Month</td></tr>
</table>
</td></tr>
<tr><td>25210</td><td>DisplayDay</td><td>N</td><td>uInt32</td><td>The valid range is 1, ..., 31. Will be set if ContractDisplayInstruction (25186) is 1 = Date or 10 = Relative Day.</td></tr>
</table>

Tag	Field Name	Req'd	Data Type	Description
25220	DisplayRelativeDay	N	int32	The valid range is -99, ..., 0, ..., 99. Negative values denote that the reference day is in the past. Will be set if *ContractDisplayInstruction (25186)* is 10 = Relative Day to the number of business days between the current business day and the reference date. It will be recalculated daily.
25212	DisplayWeek	N	uInt32	The valid range is 1, ..., 53.
25239	DisplayDayOfWeek	N	DisplayDayOfWeek (enum)	The day of week of the weekly contract. To be used with *ContractDisplayInstruction (25186)*. **Value** / **Description** 1 Monday 2 Tuesday 3 Wednesday 4 Thursday 5 Friday 6 Saturday 7 Sunday
25211	DisplayMonth	N	uInt32	The valid range is 1, ..., 12.
25189	DisplayQuarter	N	uInt32	The Display Quarter denotes the three-month period inside a year (1, ..., 4), to which a quarterly instrument refers to. The attribute is foreseen to be used for quarterly instruments.
25214	DisplaySeason	N	DisplaySeason (enum)	**SUM = Summer** if the contract month is between April and September, **WIN = Winter** if the contract month is between October and March. **Value** / **Description** SUM Summer WIN Winter
25213	DisplayYear	N	uInt32	The valid range is 2021, ..., 9999.
202	StrikePrice	N	decimal	Strike price, e.g. 52.00.
2577	StrikePricePrecision	N	uInt32	Maximum number of decimal points for the strike price. Only relevant for display purposes.
231	ContractMultiplier	N	decimal	Contract size, e.g. 100. For products with SettlMethod (1193) P = Physical, a contract multiplier with a decimal fraction is split into a deliverable integer part while the remainder is cash settled. E.g. 102.8895 is split into 102 physically delivered amount and the rest of 0.8895 is cash settled.
201	PutOrCall	N	PutOrCall (enum)	Defines if instrument is a put or call. **Value** / **Description** 0 Put 1 Call
206	OptAttribute	N	uInt32	Version of an option. The version changes as a result of corporate actions or events. The "new" option gets version zero, the "old" version gets a higher number and may exist temporarily in parallel until it is delisted. Delisting is done if no open interest exists for the Call and Put. For a new version zero a unique SecurityID is introduced.
1194	ExerciseStyle	N	ExerciseStyle (enum)	Style family of an option. **Value** / **Description** 0 European 1 American
2578	OrigStrikePrice	N	decimal	Original strike price prior to corporate action, e.g. 5.20.

Tag	Field Name	Req'd	Data Type	Description
25034	ContractGenerationNumber	N	uInt32	Contract generation.
2574	LowExercisePriceOptionIndicator	N	LepoFlag (enum)	LEPO Flag. **Value** / **Description** N – No Y – Yes
1197	ValuationMethod	N	ValuationMethod (enum)	Traditional or futures margin style. **Value** / **Description** EQTY – Premium Style FUT – Futures Style Mark-to-Market
1193	SettlMethod	N	SettlMethod (enum)	Settlement type. **Value** / **Description** C – Cash P – Physical
2579	SettlSubMethod	N	SettlSubMethod (enum)	Physical settlement type. **Value** / **Description** 1 – Shares 2 – Derivatives 3 – Payment Versus Payment 4 – Notional 5 – Cascade 99 – Other 100 – Energy
734	PriorSettlPrice	N	decimal	Previous day's settlement price. Provided in clearing notation for variance futures and total return futures instruments.
811	PriceDelta	N	decimal	Previous day's option delta provided for option instruments only.
28904	RiskSensitivityFactor	N	decimal	Instrument specific weighing factor that is applied to the notional value of an order when comparing that notional value against the remaining risk headroom during the MBRL check at order entry (nodal only)
55	Symbol	N	string	FlipId/FlexId, only for flexible instruments.
1198	ListMethod	N	ListMethod (enum)	Applicable for scaled simple instruments only. **0 = Pre-listed scaled simple security** Defines, if instrument is pre-defined by Exchange, or **1 = User requested scaled simple security** if the instrument was defined by user request. User defined instruments are deleted at the end of the day if the order book is empty. **Value** / **Description** 0 – Pre-listed only 1 – User requested
<SimpleInstrumentDescriptor> (optional) group ends				
<ComplexInstrumentDescriptorGroup> (optional) group starts				

Tag	Field Name	Req'd	Data Type	Description
893	LastFragment	Y	Last-Fragment (enum)	Indicates whether this message is the last in a sequence of messages that together convey a joint list of InstrmtLegGrp. All messages up to the last with LastFragment = Y share the same message content except InstrmtLegGrp. An application first needs to combine all InstrmtLegGrp lists before it has the complete definition of a complex instrument. *Please note that* LastFragment = N is not expected for marketplace EUREX.
28900	LegRatioMultiplier	N	uInt32	Common integer multiple of the option legs for Option Volatility Strategies.
32763	PriceNotation	N	Price-Notation (enum)	Complex instrument price notation denotes the relation between a complex instrument price and prices in the instrument legs. **Value** / **Description** 1 – Average Net Change 2 – Average
<InstrmtLegGrp> sequence starts				
555	NoLegs	N	length	
600	> LegSymbol	Y	uInt32	Product identifier of the leg security.
602	> LegSecurityID	Y	int64	Instrument identifier of the leg security.
603	> LegSecurityIDSource	Y	string	**Value** / **Description** M – Marketplace-assigned identifier
609	> LegSecurityType	Y	LegSecurity-Type (enum)	**Value** / **Description** MLEG – Part of a multi-leg instrument ULEG – Underlying leg
624	> LegSide	Y	LegSide (enum)	**Value** / **Description** 1 – Buy 2 – Sell
623	> LegRatioQty	Y	uInt32	The ratio of quantity for this individual leg relative to the entire multileg security. For details on the leg ratio rules, please refer to the T7 Functional Reference available at www.eurex.com > Support > Initiatives & Releases > T7 Release 13.1 > System Documentation > Overview and Functionality The quantity for the leg of an order can be calculated as *LegRatioQty(623)* $*$ *OrderQty(38,ETI)* $=$ the Leg quantity.
566	> LegPrice	N	decimal	Price for the leg instrument. Used for the underlying leg of a Volatility or Non-Standard Volatility Strategy.
30810	> RelatedPrice	N	decimal	This field is used to convey the price further specified by *RelatedPriceType (30419)*.
30419	> RelatedPriceType	N	Related-PriceType (enum)	Indicates what kind of price is reported in the field *RelatedPrice (30810)*. **1 = Basis** Refers to a Basis price in *RelatedPrice (30810)* **Value** / **Description** 1 – Basis

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>31194</td><td>> TradingStyle</td><td>N</td><td>Trading-Style (enum)</td><td>This field is used for all legs of a Total Return Future Strategy.
1 = Regular A Regular future instrument.
2 = TRF_TAC A Total Return Futures instrument that is traded at underlying close (TAC).
3 = TRF_TAM A Total Return Futures instrument that is traded at market (TAM).
4 = MOC A MOC (Market-On-Close) futures instrument.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Regular</td></tr><tr><td>2</td><td>TRF_TAC</td></tr><tr><td>3</td><td>TRF_TAM</td></tr><tr><td>4</td><td>MOC</td></tr></table></td></tr>
<tr><td colspan="5"><InstrmtLegGrp> sequence ends</td></tr>
<tr><td colspan="5"><ComplexInstrumentDescriptor> (optional) group ends</td></tr>
<tr><td colspan="5"><UnderlyingInstrument> (optional) group starts</td></tr>
<tr><td>31300</td><td>UnderlyingMarketSegmentID</td><td>Y</td><td>uInt32</td><td>Refers to MarketSegmentID (1300) from the underlying Product Snapshot.</td></tr>
<tr><td>309</td><td>UnderlyingSecurityID</td><td>Y</td><td>int64</td><td>Refers to SecurityID (48) from the underlying Instrument Snapshot.</td></tr>
<tr><td>305</td><td>UnderlyingSecurityID-Source</td><td>Y</td><td>string</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>M</td><td>Marketplace-assigned identifier</td></tr></table></td></tr>
<tr><td colspan="5"><UnderlyingInstrument> (optional) group ends</td></tr>
<tr><td>996</td><td>UnitOfMeasure</td><td>N</td><td>string</td><td>Used for EEX instruments.</td></tr>
<tr><td>1940</td><td>AssetType</td><td>N</td><td>Asset-Type (enum)</td><td>Used For EEX Instruments.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Emission Allowances</td></tr><tr><td>1</td><td>Other</td></tr></table></td></tr>
<tr><td>29831</td><td>AssetSubType</td><td>N</td><td>Asset-Sub-Type (enum)</td><td>Used For EEX Instruments.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>EUAE</td></tr><tr><td>1</td><td>CERE</td></tr><tr><td>2</td><td>ERUE</td></tr><tr><td>3</td><td>EUAA</td></tr><tr><td>4</td><td>Other</td></tr></table></td></tr>
<tr><td>60</td><td>TransactTime</td><td>N</td><td>timestamp</td><td>Creation time of the new flexible, scaled simple or complex security. For Instrument incremental messages (see 11.2.3, 115) with SecurityUpdateAction (980) M = Modify it is the last modification time.</td></tr>
<tr><td colspan="5"><DerivativesDescriptor> (optional) group ends</td></tr>
<tr><td colspan="5"><CashDescriptor> (optional) group starts</td></tr>
<tr><td>1787</td><td>RefTickTableID</td><td>Y</td><td>uInt32</td><td>Reference to price step table alias tick rule table identifier TickTableID (28887) from Product snapshot message.
It is also used as the reference to the default tick rule table for TES.</td></tr>
<tr><td>15</td><td>Currency</td><td>Y</td><td>string</td><td>Currency as published in ISO 4217.</td></tr>
<tr><td>120</td><td>SettlCurrency</td><td>Y</td><td>string</td><td>Settlement currency.</td></tr>
</table>

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>28890</td><td>DepositType</td><td>N</td><td>DepositType (enum)</td><td>5 = NCSC via T2S Non-collective safe custody securities to be settled via T2S (NCSC-T).<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Auslandskassenverein</td></tr><tr><td>2</td><td>Girosammelverwahrung</td></tr><tr><td>3</td><td>Streifbandverwahrung</td></tr><tr><td>4</td><td>Wertpapierrechnung</td></tr><tr><td>5</td><td>NCSC via T2S</td></tr></table></td></tr>
<tr><td>225</td><td>IssueDate</td><td>N</td><td>uInt32</td><td>Issue date of instrument.</td></tr>
<tr><td>561</td><td>RoundLot</td><td>N</td><td>decimal</td><td>Minimum Tradable Unit.</td></tr>
<tr><td>562</td><td>MinTradeVol</td><td>N</td><td>decimal</td><td>Minimum Order Quantity.</td></tr>
<tr><td>1140</td><td>MaxTradeVol</td><td>N</td><td>decimal</td><td>This field contains the maximum quantity of a regular order or quote, which is allowed to be traded in a given instrument. [22]</td></tr>
<tr><td>28804</td><td>MaxTradeVal</td><td>N</td><td>uInt64</td><td>This field contains the maximum value of a regular order or quote, which is allowed to be traded in a given instrument. [23]</td></tr>
<tr><td>30341</td><td>QuotingStartTime</td><td>N</td><td>string</td><td>Quoting Period Start Time (HH:mm:ss).</td></tr>
<tr><td>30345</td><td>QuotingEndTime</td><td>N</td><td>string</td><td>Quoting Period End Time (HH:mm:ss).</td></tr>
<tr><td>31803</td><td>InstrumentAuctionType</td><td>N</td><td>Instrument-Auction-Type (enum)</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Default</td></tr><tr><td>1</td><td>Single Auction</td></tr><tr><td>2</td><td>Special Auction</td></tr></table></td></tr>
<tr><td>28913</td><td>MidpointTrading</td><td>N</td><td>Midpoint-Trading (enum)</td><td>Indicates whether the instrument is enabled for Midpoint trading.<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>28914</td><td>MidpointExecVenueID</td><td>N</td><td>string</td><td>Market Identifier Code (ISO 10383) used for reporting midpoint trades to the supervisory authority.</td></tr>
<tr><td colspan="5"><InstrumentParties> (optional) sequence starts</td></tr>
<tr><td>1018</td><td>NoInstrumentParties</td><td>Y</td><td>length</td><td>Number of parties.</td></tr>
<tr><td>1019</td><td>> InstrumentPartyID</td><td>Y</td><td>string</td><td>Identifies a party associated with an instrument.</td></tr>
<tr><td>1050</td><td>> InstrumentPartyIDSource</td><td>Y</td><td>Instrument-PartyId-SourceType (enum)</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>D</td><td>Proprietary</td></tr></table></td></tr>
<tr><td>1051</td><td>> InstrumentPartyRole</td><td>Y</td><td>Instrument-PartyRole-Type (enum)</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>21</td><td>Clearing Organization</td></tr><tr><td>66</td><td>Market Maker</td></tr></table></td></tr>
<tr><td>2378</td><td>> InstrumentPartyRole-Qualifier</td><td>N</td><td>Instrument-PartyRole-Qualifier-Type (enum)</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>20</td><td>Designated Sponsor</td></tr><tr><td>21</td><td>Specialist</td></tr></table></td></tr>
<tr><td colspan="5"><InstrumentParties> (optional) sequence ends</td></tr>
<tr><td colspan="5"><BondsAndWarrantsGroup> (optional) group start</td></tr>
</table>

[22] In the Cash Market Instrument Reference Data Guide the field name is *Maximum Order Quantity*
[23] In the Cash Market Instrument Reference Data Guide the field name is *Maximum Order Value*

Tag	Field Name	Req'd	Data Type	Description
311	UnderlyingSymbol	N	string	Underlying symbol.
223	CouponRate	N	decimal	The coupon rate of the respective Bond.
28895	PreviousCouponPaymentDate	N	uInt32	Previous Coupon payment date.
224	CouponPaymentDate	N	uInt32	Upcoming Coupon payment date.
1950	CouponDayCount	N	AccruedInterestCalculationMethod (enum)	Defines the Accrued interest Calculation Method. **Value** / **Description** 1 – 30/360 3 – 30/360M 6 – Act/360 7 – Act/365 (Fixed) 8 – Act/Act (AFB) 9 – Act/Act (ICMA) 11 – Act/Act (ISDA) 14 – Act/365L
1946	CouponType	N	CouponType (enum)	Is set to **1 = Zero** when the Bond has no coupons. **Value** / **Description** 1 – Zero 2 – FixedRate 3 – FloatingRate 4 – Structured
470	CountryOfIssue	N	string	ISO Country code. The calculated accrued interest rate is rounded to the 12th decimal, except for the following country codes: **Value** / **Description** FR – 9th decimal IT – 7th decimal PL – 7th decimal HU – 7th decimal
25170	FlatIndicator	N	FlatIndicator (enum)	The Flat Indicator of a bond. **Value** / **Description** 1 – No Flat 2 – Flat 3 – X-Flat
30762	WarrantType	N	WarrantType (enum)	Specific warrant type **Value** / **Description** 1 – Call 2 – Put 3 – Range 4 – Certificate 5 – Other
25200	CoverIndicator	N	CoverIndicator (enum)	Cover Indicator **Value** / **Description** 1 – Intraday Cover 2 – Longterm Cover 3 – Activated Cover
<BondsAndWarrantsGroup> (optional) group ends				

Tag	Field Name	Req'd	Data Type	Description
<VolatilityCorridorGroup> (optional) group starts				
25221	VolatilityCorridorOpeningAuction	N	uInt32	Reference to Volatility Corridor Table for Opening Auction. The mapping of integer values to the corresponding corridor table is available at: www.xetra.com > Instruments > All tradable instruments
25222	VolatilityCorridorIntradayAuction	N	uInt32	Reference to Volatility Corridor Table for Intraday Auction. The mapping of integer values to the corresponding corridor table is available at: www.xetra.com > Instruments > All tradable instruments
25223	VolatilityCorridorClosingAuction	N	uInt32	Reference to Volatility Corridor Table for Closing Auction. The mapping of integer values to the corresponding corridor table is available at: www.xetra.com > Instruments > All tradable instruments
25224	VolatilityCorridorContinuous	N	uInt32	Reference to Volatility Corridor Table in Continuous. The mapping of integer values to the corresponding corridor table is available at: www.xetra.com > Instruments > All tradable instruments
<VolatilityCorridorGroup> (optional) group ends				
<SecurityClassificationGrp> (optional) sequence starts				
1582	NoSecurityClassifications	Y	length	Always 1
1583	> SecurityClassificationReason	Y	SecurityClassificationReasonType (enum)	Always 3 **Value** / **Description**: 3 – Entitlement / Eligibility
1584	> SecurityClassificationValue	Y	SecurityClassificationValueType (enum)	**Value** / **Description**: 1 – Eligible for all trading activities; 2 – Not eligible for OTC Trade Upload; 3 – Eligible only for OTC Trade Upload
<SecurityClassificationGrp> (optional) sequence ends				
<CashDescriptor> (optional) group ends				
<InstrumentAttributes> (optional) sequence starts				
870	NoInstrAttrib	Y	length	Number of instrument attributes.

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>871</td><td>> InstrAttribType</td><td>Y</td><td>Instrument-Attribute-Type (enum)</td><td>111 = Pre Trade LIS Value This parameter was first introduced with T7 Release 7.0. On TES entry, TES trade price multiplied by the (side) quantity has to be at least equal to or higher than Pre Trade LIS Value. Depending on the difference to this value, the delay for the TES trade disclosure will be determined.

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>100</td><td>Minimum Reserve Order Volume</td></tr>
<tr><td>101</td><td>Minimum Display Volume</td></tr>
<tr><td>102</td><td>Issuer Name</td></tr>
<tr><td>103</td><td>Issuer Number</td></tr>
<tr><td>104</td><td>Market Type</td></tr>
<tr><td>105</td><td>Market Type Supplement</td></tr>
<tr><td>106</td><td>Reporting Market</td></tr>
<tr><td>107</td><td>Cum-Ex Indicator</td></tr>
<tr><td>108</td><td>Product Assignment Group</td></tr>
<tr><td>109</td><td>Product Assignment Group Description</td></tr>
<tr><td>110</td><td>Domestic Indicator</td></tr>
<tr><td>111</td><td>Pre Trade LIS Value</td></tr>
<tr><td>112</td><td>Illiquid As Defined By Exchange</td></tr>
<tr><td>113</td><td>Market Making Obligation</td></tr>
<tr><td>114</td><td>Liquid As Defined By Regulator</td></tr>
<tr><td>115</td><td>Eligible For Stressed Market Conditions</td></tr>
<tr><td>117</td><td>Multi CCP-eligibility</td></tr>
<tr><td>118</td><td>Pool Factor</td></tr>
<tr><td>119</td><td>Indexation Coefficient</td></tr>
<tr><td>120</td><td>Trading On Terms Of Issue</td></tr>
<tr><td>121</td><td>Issuer Business Unit</td></tr>
<tr><td>122</td><td>Allow Knock out</td></tr>
<tr><td>123</td><td>Has PLP</td></tr>
<tr><td>124</td><td>PLP Deferral Time</td></tr>
<tr><td>125</td><td>Warrant Strike Price</td></tr>
<tr><td>126</td><td>Reporting Market TES</td></tr>
<tr><td>127</td><td>Liquidity Provider User Group</td></tr>
<tr><td>128</td><td>Specialist User Group</td></tr>
<tr><td>129</td><td>Liquidity Class</td></tr>
</table></td></tr>
<tr><td>872</td><td>> InstrAttribValue</td><td>Y</td><td>string</td><td></td></tr>
<tr><td colspan="5"><InstrumentAttributes> (optional) sequence ends</td></tr>
<tr><td colspan="5"><Events> (optional) group starts</td></tr>
<tr><td>864</td><td>NoEvents</td><td>N</td><td>length</td><td>Number of events.</td></tr>
</table>

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>865</td><td>> EventType</td><td>Y</td><td>EventType (enum)</td><td>115 = Final Settlement Reference Date will only be set for derivatives market products. Identifies the day, on which the underlying information is taken that is the basis of the determination of the final settlement price at the expiration of the instrument.<table><tr><th>Value</th><th>Description</th></tr><tr><td>7</td><td>Last Eligible Trade Date</td></tr><tr><td>100</td><td>First Eligible Trade Date</td></tr><tr><td>101</td><td>Capital Adjustment Date</td></tr><tr><td>102</td><td>Dividend Payment Date</td></tr><tr><td>115</td><td>Final Settlement Reference Date</td></tr></table></td></tr>
<tr><td>866</td><td>> EventDate</td><td>Y</td><td>uInt32</td><td></td></tr>
<tr><td colspan="5"><Events> (optional) group ends</td></tr>
<tr><td>2576</td><td>InstrumentPricePrecision</td><td>N</td><td>uInt32</td><td>Display decimals.</td></tr>
<tr><td>969</td><td>MinPriceIncrement</td><td>N</td><td>decimal</td><td>Defines the minimum increment for trade prices (tick size). This value is identical for all simple instruments of a product. Trading and clearing tick sizes, i.e. increments for trade prices in trading and clearing notation are the same unless MinPriceIncrementClearing(28888) is present.</td></tr>
<tr><td>28888</td><td>MinPriceIncrementClearing</td><td>N</td><td>decimal</td><td>Defines the minimum increment for trade prices in clearing notation (clearing tick size). Used for products to distinguish between tick sizes for trading and clearing notation, e.g. Total Return Futures.</td></tr>
<tr><td>1146</td><td>MinPriceIncrementAmount</td><td>N</td><td>decimal</td><td>Defines the minimum price movement in the respective currency (tick value). This value is identical for all instruments of a product. Amounts are only relevant for clearing tick sizes, i.e. to be multiplied with MinPriceIncrement(969) unless MinPriceIncrementClearing(28888) is present.</td></tr>
<tr><td>541</td><td>MaturityDate</td><td>N</td><td>uInt32</td><td>Expiration day of the instrument (YYYYMMDD).</td></tr>
<tr><td>200</td><td>MaturityMonthYear</td><td>N</td><td>uInt32</td><td>Expiration month (YYYYMM). Note that the actual expiration day provided by MaturityDate (541) may be in a different month.</td></tr>
<tr><td>762</td><td>SecuritySubType</td><td>N</td><td>uInt32</td><td>Standard strategy type for complex instruments.
The mapping of integer values to the corresponding strategy types is available at:
www.eurex.com > Data > Trading files > Product parameter files > Trading parameters

Bond type is published as SecuritySubType.
The mapping of integer values to the corresponding bond types is available at:
www.xetra.com > Instruments > All tradable instruments</td></tr>
<tr><td colspan="5"><RelatedInstrumentGrp> (optional) group starts</td></tr>
<tr><td>1650</td><td>RelatedSecurityID</td><td>Y</td><td>int64</td><td>The instrument identifier uniquely identifies the related instrument.</td></tr>
<tr><td>1651</td><td>RelatedSecurityIDSource</td><td>Y</td><td>string</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>M</td><td>Marketplace-assigned identifier</td></tr></table></td></tr>
<tr><td colspan="5"><RelatedInstrumentGrp> (optional) group end</td></tr>
<tr><td colspan="5"><MarketSegmentGrp> sequence starts</td></tr>
<tr><td>1310</td><td>NoMarketSegments</td><td>Y</td><td>length</td><td>always “1”.</td></tr>
<tr><td>1300</td><td>> MarketSegmentID</td><td>Y</td><td>uInt32</td><td>Product identifier or product pool identifier, e.g. 89.</td></tr>
</table>

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>1144</td><td>> ImpliedMarketIndicator</td><td>N</td><td>Implied-Market-Indicator (enum)</td><td>Defines the matching method. Applicable for derivatives market instruments only.<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Not implied</td></tr><tr><td>3</td><td>Both implied-in and implied-out</td></tr></table></td></tr>
<tr><td>1377</td><td>> MultilegModel</td><td>N</td><td>Multileg-Model (enum)</td><td>Applicable for derivatives market instruments only.
0 = Pre-defined multileg security Defines, if instrument is predefined by Exchange, or
1 = User-defined multileg security if the instrument was defined by user request. User defined instruments are deleted at the end of the day if the order book is empty.<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Pre-defined multileg security</td></tr><tr><td>1</td><td>User-defined multileg security</td></tr></table></td></tr>
<tr><td>423</td><td>> PriceType</td><td>N</td><td>PriceType (enum)</td><td>The unit in which an instrument is quoted/stated when buying or selling.
2 = Shares Shares for cash market products, Monetary for derivatives market products.<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Percent</td></tr><tr><td>2</td><td>Shares</td></tr><tr><td>22</td><td>Points</td></tr></table></td></tr>
<tr><td>28876</td><td>> PostTradeAnonymity</td><td>N</td><td>PostTrade-Anonymity-Type (enum)</td><td>Only applicable for cash market instruments.<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr><tr><td>2</td><td>Central Counterparty</td></tr></table></td></tr>
<tr><td>28878</td><td>> SettlBusinessDays</td><td>N</td><td>uInt32</td><td>Indicates the number of business days from trade execution after which settlement is to be effected. Applicable for cash market instruments only.</td></tr>
<tr><td colspan="5"><QuoteSizeRules> (optional) sequence starts</td></tr>
<tr><td>2558</td><td>> NoQuoteSizeRules</td><td>N</td><td>length</td><td>Applicable for cash market instruments only. Only one table.</td></tr>
<tr><td>647</td><td>> > MinBidSize</td><td>Y</td><td>decimal</td><td>Bid side minimum quote quantity.</td></tr>
<tr><td>648</td><td>> > MinOfferSize</td><td>Y</td><td>decimal</td><td>Offer side minimum quote quantity.</td></tr>
<tr><td colspan="5"><QuoteSizeRules> (optional) sequence ends</td></tr>
<tr><td colspan="5"><PriceRangeRules> (optional) sequence starts</td></tr>
<tr><td>2550</td><td>> NoPriceRangeRules</td><td>Y</td><td>length</td><td>Only one table.</td></tr>
<tr><td>2556</td><td>> > PriceRangeRuleID</td><td>Y</td><td>uInt32</td><td>Reference to table identifier from product level messages.</td></tr>
<tr><td colspan="5"><PriceRangeRules> (optional) sequence ends</td></tr>
<tr><td colspan="5"><TradingSessionRules> (optional) sequence starts</td></tr>
<tr><td>1309</td><td>> NoTradingSessionRules</td><td>N</td><td>length</td><td>Applicable for cash market instruments only. Only one table.</td></tr>
<tr><td>336</td><td>> > TradingSessionID</td><td>Y</td><td>Trading-SessionID (enum)</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Day</td></tr></table></td></tr>
</table>

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>625</td><td>> > TradingSessionSubID</td><td>Y</td><td>Trading-Session-SubID (enum)</td><td>3 = Continuous is used for Trading Model Continuous Trading
6 = Scheduled Intraday Auction is used for trading model One Auction
8 = Any Auction is used for trading model Multiple Auctions
<table><tr><th>Value</th><th>Description</th></tr><tr><td>3</td><td>Continuous</td></tr><tr><td>6</td><td>Scheduled Intraday Auction</td></tr><tr><td>8</td><td>Any Auction</td></tr><tr><td>103</td><td>Continuous Auction Issuer</td></tr><tr><td>104</td><td>Continuous Auction Specialist</td></tr></table></td></tr>
<tr><td colspan="5"><TradingSessionRules> (optional) sequence ends</td></tr>
<tr><td colspan="5"><MarketSegmentGrp> sequence ends</td></tr>
</table>

11.2.3 Instrument incremental message

Delivered on: T7 RDI incremental feed, T7 RDI snapshot feed

The *Instrument incremental* message is used in derivatives markets to report complex, (scaled simple) and flexible instruments. For a complex instrument, the *LegSecurityID (602)* is the *SecurityID (48)* of the instrument that is part (leg) of the complex instrument.
In cash markets *Instrument incremental* messages are used to report the activation and attribute change of instruments setup as covers.

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>35</td><td>MsgType</td><td>Y</td><td>string</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>BP</td><td>Security Definition Update Report</td></tr></table></td></tr>
<tr><td>980</td><td>SecurityUpdateAction</td><td>Y</td><td>Security-Update-Action (enum)</td><td>A = Add Derivatives: Add of complex or flexible instruments.
M = Modify Derivatives: Intraday update of complex instruments (expected during start of day), e.g. SecurityDesc (107) or IsPrimary (25216) can change during roll over.
Cash: Intraday activation and attribute change of covers.<table><tr><th>Value</th><th>Description</th></tr><tr><td>A</td><td>Add</td></tr><tr><td>M</td><td>Modify</td></tr></table></td></tr>
</table>

Note: A subset of the fields from the Instrument snapshot message are also part of the message.

11.2.4 Variance futures status message

Delivered on: T7 RDI incremental feed, T7 RDI snapshot feed

The *Variance Futures Status* message is used to convey information specific to variance futures instruments.

During a normal day, a first initial snapshot message with the "Preliminary" attributes is published at the start of day, completed later by only one new message with the "Final" values.

In case the current business day is a holiday for the product, the status message will contain the information of the previous trading day.

Tag	Field Name	Req'd	Data Type	Description
35	MsgType	Y	string	**Value** / **Description**: f – Security Status
34	MsgSeqNum	Y	uInt32	
1300	MarketSegmentID	Y	uInt32	Product identifier.
48	SecurityID	Y	int64	The instrument identifier uniquely identifies an instrument.
22	SecurityIDSource	Y	string	**Value** / **Description**: M – Marketplace-assigned identifier
167	SecurityType	Y	string	**Value** / **Description**: VAR – Variance Futures
<ClearingPriceParameters> sequence starts				
2580	NoClearingPriceParameters	Y	length	One table for each type of business day. Always “2“.
2581	> BusinessDayType	Y	Business-Day-Type (enum)	**Value** / **Description**: 5 – Preceding Day; 1 – Current Day
2582	> ClearingPriceOffset	N	decimal	Price constant defined on the instrument level and used for the clearing price conversion.
2583	> VegaMultiplier	N	uInt64	Constant multiplier of the Notional Vega defined on the product level and used for the clearing quantity conversion.
2584	> AnnualTradingBusinessDays	N	uInt32	Approximate number of trading days during one year defined as a constant on the product level and used for the calculation of RealizedVariance (2587).
2585	> TotalTradingBusinessDays	N	uInt32	Total number of trading days of the instrument, including the first and the last trading day, which is one day before the expiration.
2586	> TradingBusinessDays	N	uInt32	Total number of trading days already passed since the introduction of the instrument.
2588	> StandardVariance	N	decimal	Used as a variance reference for the trading price conversion and the settlement price calculation. The standard variance is set to the implied volatility at the end of the first trading day. It is provided during the first trading day when parameters are final and then remains unchanged for the rest of the trading days.
2589	> RelatedClosePrice	N	decimal	Closing price of the underlying on the product level and used to calculate the realised variance. Also provided for previous day.
2587	> RealizedVariance	N	decimal	Calculated from all underlying closing prices since the introduction of the instrument adjusted by AnnualTradingBusinessDays (2584). Also provided for previous day.

Tag	Field Name	Req'd	Data Type	Description
1188	> Volatility	N	decimal	Implied volatility that has been used to calculate ClearingSettlPrice (2528). On the first trading day it is the base for StandardVariance (2588) of the instrument. Represents trading notation of ClearingSettlPrice (2528) and is only provided for previous day.
2528	> ClearingSettlPrice	N	decimal	Settlement price in clearing notation. Only provided for previous day.
2592	> CalculationMethod	N	Calculation-Method-Type (enum)	Indicates whether the automatic calculation of the parameters has been disabled and parameters have been manually defined. Intraday update is possible at any time. **Value** / **Description** 0 – Automatic 1 – Manual
<ClearingPriceParameters> sequence ends				

Notes:
On the T7 RDI incremental feed: The *MsgSeqNum* (34) starts with 1 and increments by 1 for each message.
On the T7 RDI snapshot feed: The *MsgSeqNum* (34) is contiguous across all messages regardless of message type. See figure 9 chapter 9.2.1, General structure of the snapshot cycle.

11.2.5 Total return futures status message

Delivered on: T7 RDI incremental feed, T7 RDI snapshot feed

The *Total Return Futures Status* message is used to convey information specific to total return futures instruments.

During a normal day, a first message is published at the start of day with the following initial attributes: *AnnualCalendarDays* (28881), *RemainingCalendarDays* (28880) for the current day (*BusinessDayType* (2581) 1=Current Day) and *PriorSettlPrice* (734).

Later, still at the beginning of the day, a new message is sent containing the initial attributes completed with the following preliminary attributes: *OvernightInterestRate* (2590), *CurrentPaymentParameter* (28883), *AccruedPaymentParameter* (28884), *RelatedIndexValue* (28882), *CurrentCollectionParameter* (28885), *AccruedCollectionParameter* (28886) for the current day and *RelatedClosePrice* (2589) for the previous day (*BusinessDayType* (2581) 5=Preceding Day).

Finally, at the end of the day, a final message is sent with the final attributes, *RelatedClosePrice* (2589) for the current day and *SettlPrice* (730).

No conversion parameters will be created for a business day that is not a trading day for the corresponding product.

Tag	Field Name	Req'd	Data Type	Description
35	MsgType	Y	string	**Value** / **Description**: f – Security Status
34	MsgSeqNum	Y	uInt32	
1300	MarketSegmentID	Y	uInt32	Product identifier.
48	SecurityID	Y	int64	The instrument identifier uniquely identifies an instrument.
22	SecurityIDSource	Y	string	**Value** / **Description**: M – Marketplace-assigned identifier
167	SecurityType	Y	string	**Value** / **Description**: TRF – Total Return Futures
734	PriorSettlPrice	Y	decimal	Daily settlement price of the previous day in trading notation (TRF spread).
730	SettlPrice	N	decimal	Daily settlement price of the current day in trading notation (TRF spread).
<ClearingPriceParameters> sequence starts				
2580	NoClearingPriceParameters	Y	length	Two tables for previous and current day. Always "2".
2581	> BusinessDayType	Y	BusinessDayType (enum)	**Value** / **Description**: 5 – Preceding Day; 1 – Current Day
28881	> AnnualCalendarDays	N	uInt32	Annualisation Factor, represents the number of calendar days during one year as applied in the calculations.
28880	> RemainingCalendarDays	N	int32	Days to maturity, number of calendar days between the current day settlement date and the instrument expiration settlement date. Used for the trade price conversion from trading to clearing notation.
2590	> OvernightInterestRate	N	decimal	Funding rate, used for the funding calculation of the current day. Represents the overnight or the periodic interest rate established on the previous day. Becomes available in the morning before trading starts.

Tag	Field Name	Req'd	Data Type	Description
28883	> CurrentPaymentParameter	N	decimal	Daily funding, calculated from the funding rate applied to the previous day underlying close price.
28884	> AccruedPaymentParameter	N	decimal	Accrued funding, accumulated from the daily fundings since the product launch. Used for the trade price conversion from trading to clearing notation.
28882	> RelatedIndexValue	N	decimal	Distribution index, defined for the current day. Becomes available in the morning before trading starts.
28885	> CurrentCollectionParameter	N	decimal	Daily distribution, calculated from the difference between the current and the previous day distribution index.
28886	> AccruedCollectionParameter	N	decimal	Accrued distribution, accumulated from the daily distributions since the product launch. Used for the trade price conversion from trading to clearing notation.
30159	> CurrentAccruedInterestAmt	N	decimal	Coupon payment, for which the coupon period is ongoing, multiplied by the ratio of the already passed time of the coupon period and the entire coupon period.
2589	> RelatedClosePrice	N	decimal	For BusinessDayType (2581) **5 = Preceding Day** The previous day's underlying close price. It is used as preliminary underlying price for the trade price conversion from trading to clearing notation, before the current day's final underlying price is known. For BusinessDayType (2581) **1 = Current Day** The current day's final underlying price for the trade price conversion from trading to clearing notation. For most TRF products, it is the underlying's close price of the current day.
28909	> ConversionMode	N	ConversionMode (enum)	This attribute allows to distinguish between the different modes regarding the trade price conversion from trading notation to clearing notation. **Value** / **Description**: 0 Initial; 1 Preliminary; 2 Final
28910	> ConversionModeQualifier	N	ConversionModeQualifier (enum)	This attribute allows to distinguish between an early and a late preliminary conversion mode, for TRF products with an intermediate TRF parameter maintenance. **Value** / **Description**: 1 Early; 2 Late
<ClearingPriceParameters> sequence ends				

Notes:
On the T7 RDI incremental feed: The *MsgSeqNum* (34) starts with 1 and increments by 1 for each message.
On the T7 RDI snapshot feed: The *MsgSeqNum* (34) is contiguous across all messages regardless of message type. See figure 9 chapter 9.2.1, General structure of the snapshot cycle.

11.2.6 Trade At Reference Price status message

Delivered on: T7 RDI incremental feed, T7 RDI snapshot feed

The *Trade At Reference Price Status* message is used to convey information specific to trade at reference price instruments.

During a normal day, a first message is published at the start of day with the following initial attributes: *ClearingPriceOffset* (2582) for the current day and *RelatedClosePrice* (2589) for the previous day (*BusinessDayType* (2581) = Preceding Day). At the end of the day, a final message is sent with the attribute *RelatedClosePrice* (2589) ()*BusinessDayType* (2581) = Current Day).

Tag	Field Name	Req'd	Data Type	Description
35	MsgType	Y	string	**Value** / **Description**: f – Security Status
34	MsgSeqNum	Y	uInt32	
1300	MarketSegmentID	Y	uInt32	Product identifier.
48	SecurityID	Y	int64	The instrument identifier uniquely identifies an instrument.
22	SecurityIDSource	Y	string	**Value** / **Description**: M – Marketplace-assigned identifier
167	SecurityType	Y	string	**Value** / **Description**: TARP – Trade At Reference Price
<ClearingPriceParameters> sequence starts				
2580	NoClearingPriceParameters	Y	length	Two tables for previous and current day. Always "2".
2581	> BusinessDayType	Y	Business-Day-Type (enum)	**Value** / **Description**: 5 – Preceding Day; 1 – Current Day
2582	> ClearingPriceOffset	N	uInt32	Price constant defined on the product level and used for the clearing price conversion.
2589	> RelatedClosePrice	N	decimal	The preliminary reference price is published with BusinessDayType (2581) set to Preceding Day, although it may change on the current day. The final reference price is published with BusinessDayType (2581) set to Current Day.
<ClearingPriceParameters> sequence ends				

Notes:
On the T7 RDI incremental feed: The *MsgSeqNum* (34) starts with 1 and increments by 1 for each message.
On the T7 RDI snapshot feed: The *MsgSeqNum* (34) is contiguous across all messages regardless of message type. See figure 9 chapter 9.2.1, General structure of the snapshot cycle.

11.3 Market data messages

The market data feeds disperse public market data via the T7 EMDI and the T7 MDI.

Public market data for all instruments are distributed over preconfigured multicast addresses. It is possible to configure multiple instruments over one multicast address and the depth of information to be disseminated can be configured on a per product basis. The multicast address and port combinations are different for the T7 EMDI and the T7 MDI.

Two different messages are used for order book updates: The *depth incremental* is sent if the order book changes (driven by an order book event). Conversely, the *depth snapshot* is sent in certain intervals independent from any change in the order book (time driven).

The message layout for the T7 EMDI and T7 MDI is the same.

11.3.1 Depth snapshot message

Delivered on: T7 EMDI snapshot feed, T7 MDI data feed

This message provides periodic updates for orders and trades independent from any change of the order book. Updates are available up to the maximum depth defined by the exchange in the field *MarketDepth (264)*. The Snapshot can be synchronized with the incremental message as described in chapter 6.5, Update the order book. One message per instrument with pre- and post trade data is sent. An empty book is disseminated during the product states as indicated in chapter 9.3, General order book rules and mechanics, bullet 5.

Tag	Field Name	Req'd	Data Type	Description
35	MsgType	Y	string	**Value** / **Description**: W – Market Data Snapshot Full Refresh
34	MsgSeqNum	N	uInt32	Not used by unnetted feed (EMDI) where field is never present. The sequence number of the message is incremented per product across all message types.
49	SenderCompID	Y	uInt32	Unique id of a sender.
369	LastMsgSeqNumProcessed	N	uInt32	Not used by netted feed (MDI) where field is never present. Last message sequence number sent regardless of message type.
1187	RefreshIndicator	N	Refresh-Indicator (enum)	Used by netted feed (MDI) only. If set then the depth snapshot information has not been sent with the depth incremental before. **Value** / **Description**: Y – Mandatory Refresh; N – Optional Refresh
1300	MarketSegmentID	Y	uInt32	Product identifier, e.g. "89".
48	SecurityID	Y	int64	Instrument identifier, e.g. "8852".
22	SecurityIDSource	Y	string	Source Identification. **Value** / **Description**: M – Marketplace-assigned Identifier

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>1227</td><td>ProductComplex</td><td>Y</td><td>Product-Complex (enum)</td><td>Type of instrument<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Simple Instrument</td></tr><tr><td>2</td><td>Standard Option Strategy</td></tr><tr><td>3</td><td>Non-Standard Option Strategy</td></tr><tr><td>4</td><td>Volatility Strategy</td></tr><tr><td>5</td><td>Futures Spread</td></tr><tr><td>6</td><td>Inter-Product Spread</td></tr><tr><td>7</td><td>Standard Futures Strategy</td></tr><tr><td>8</td><td>Pack and Bundle</td></tr><tr><td>9</td><td>Strip</td></tr><tr><td>10</td><td>Flexible Instrument</td></tr><tr><td>11</td><td>Commodity Strips</td></tr><tr><td>12</td><td>Scaled Simple Instrument</td></tr><tr><td>13</td><td>Non-Standard Volatility Strategy</td></tr><tr><td>14</td><td>Total Return Future Strategy</td></tr></table></td></tr>
<tr><td>965</td><td>SecurityStatus</td><td>Y</td><td>Security-Status (enum)</td><td>Status of the instrument.
2 = Inactive will be set for pending deletions of complex instruments.
4 = Expired will be set for instruments that have expired intraday.
2 = Inactive and 4 = Expired are only applicable for derivatives market products.
11 = Pending Deletion is used for cash instruments when the last trading date is exceeded and the instrument will be deleted soon. Those instruments might become 1 = Active again intraday via Market Data intraday State Change messages.
6 = Knocked-Out, 7 = Knock-Out Revoked and 12 = Knocked-Out And Suspended are only applicable in trading model Continuous Auction Issuer.<table><tr><th>Value</th><th>Description</th></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>6</td><td>Knocked-Out</td></tr><tr><td>7</td><td>Knock-Out Revoked</td></tr><tr><td>9</td><td>Suspended</td></tr><tr><td>11</td><td>Pending Deletion</td></tr><tr><td>12</td><td>Knocked-Out And Suspended</td></tr></table></td></tr>
<tr><td>25045</td><td>TESSecurityStatus</td><td>N</td><td>Security-Status (enum)</td><td>Status of the instrument for TES trading.<table><tr><th>Value</th><th>Description</th></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>

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>779</td><td>LastUpdateTime</td><td>Y</td><td>timestamp</td><td>Time of last change for SecurityID (nanoseconds).
This can be any trade, change of the orderbook on any price level, or also a product or instrument state change information conveyed in this message.</td></tr>
<tr><td colspan="5"><MDSshGrp> sequence starts</td></tr>
<tr><td>268</td><td>NoMDEntries</td><td>Y</td><td>length</td><td></td></tr>
<tr><td>1024</td><td>> MDOriginType</td><td>Y</td><td>MDOrigin-Type (enum)</td><td>0 = Book is for on-exchange trading.
1 = Off-Book is for TES trading only. Only applicable for MDEntryType 2 = Trade or B = Trade Volume.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Book</td></tr><tr><td>1</td><td>Off-Book</td></tr></table></td></tr>
<tr><td>269</td><td>> MDEntryType</td><td>Y</td><td>MDEntry-Type (enum)</td><td>Q = Auction Clearing Price is sent as indicative information during the auction.
J = Empty Book is sent during product states “Start-Of-Day” and “Pre-Trading” or when no price levels exist. During “Post-Trading“ and “End-Of-Day“ ToB prices are distributed.
B = Trade Volume The total traded volume of units traded during the day can be found in the MDEntrySize field. Please note that the total traded volume may include coherent volume (from direct matching of complex instruments) as well.
b = Market Bid, c = Market Offer Applicable for cash market products only.
A = Imbalance Applicable for Continuous Auction Issuer trading model only. Is used instead of QuoteCondition Z = Order Imbalance for products for which the imbalance side is not disclosed.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Bid</td></tr><tr><td>1</td><td>Offer</td></tr><tr><td>2</td><td>Trade</td></tr><tr><td>J</td><td>Empty Book</td></tr><tr><td>Q</td><td>Auction Clearing Price</td></tr><tr><td>B</td><td>Trade Volume</td></tr><tr><td>b</td><td>Market Bid</td></tr><tr><td>c</td><td>Market Offer</td></tr><tr><td>A</td><td>Imbalance</td></tr></table></td></tr>
<tr><td>1021</td><td>> MDBookType</td><td>N</td><td>MDBook-Type (enum)</td><td>Price depth information or top-of-book information.
Top-of-book is only applicable to IPS, i.e. requires the presence of MDSubBookType.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Top of Book</td></tr><tr><td>2</td><td>Price Depth</td></tr></table></td></tr>
</table>

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>1173</td><td>> MDSubBookType</td><td>N</td><td>MDSubBook-Type (enum)</td><td>Optional qualifier to MDBookType. For IPS only.
1 = IPS implied volume without quantity restriction Synthetically matchable quantity resulting from IPS instruments that is not subject to quantity restrictions.
2 = IPS implied volume with quantity restriction Synthetically matchable quantity resulting from IPS instruments that is subject to quantity restrictions.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>IPS implied volume without quantity restriction</td></tr><tr><td>2</td><td>IPS implied volume with quantity restriction</td></tr></table></td></tr>
<tr><td>828</td><td>> TrdType</td><td>N</td><td>TrdType [24] (enum)</td><td>For MDOriginType 0 = Book defines when the trade happens. Only present for MDEntryType 2 = Trade and TradeCondition AW = Last Auction Price.
1, 2, 12, 50, 54, 55, 1000, 1001, 1002, 1004, 1006, 1007 and 1017 only present for MDEntryType 2 = Trade and MDOriginType 1 = Off-Book.
1 = Block Trade Used for derivatives and cash market products. For cash market products used to report T7 Entry Service (TES) trades of TES Type Large in Scale (LIS).
1107 = IPO Auction Trade Applicable for cash market products only.
54 = OTC Applicable for T7 Entry Service (TES) for cash market products only.
1006 = Xetra / Eurex Enlight triggered Trade may be set together with MDOriginType (1024) 1 = Off-Book and TradeCondition (277) U = Exchange Last.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Block Trade</td></tr><tr><td>2</td><td>Exchange For Physical (EFP)</td></tr><tr><td>12</td><td>Exchange For Swap (EFS)</td></tr><tr><td>54</td><td>OTC</td></tr><tr><td>55</td><td>Exchange Basis Facility</td></tr><tr><td>1000</td><td>Vola Trade</td></tr><tr><td>1001</td><td>EFP-Fin Trade</td></tr><tr><td>1002</td><td>EFP-Index-Futures Trade</td></tr><tr><td>1004</td><td>Block Trade at Market</td></tr><tr><td>1006</td><td>Xetra / Eurex Enlight triggered Trade</td></tr><tr><td>1007</td><td>Block QTPIP Trade</td></tr><tr><td>1017</td><td>Delta Trade At Market</td></tr><tr><td>1100</td><td>Opening Auction Trade</td></tr><tr><td>1101</td><td>Intraday Auction Trade</td></tr><tr><td>1102</td><td>Volatility Auction Trade</td></tr><tr><td>1103</td><td>Closing Auction Trade</td></tr><tr><td>1107</td><td>IPO Auction Trade</td></tr><tr><td>1108</td><td>Liquidity Improvement Cross</td></tr></table></td></tr>
</table>

[24] Cross Auction Trade is never disseminated in the TrdType field of the Depth Snapshot message. However, it is disseminated in the TrdType field of the Depth Incremental message

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>336</td><td>> TradingSessionID</td><td>N</td><td>Trading-SessionID (enum)</td><td>Always attached to the first MDEntry.<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Day</td></tr><tr><td>3</td><td>Morning</td></tr><tr><td>5</td><td>Evening</td></tr><tr><td>6</td><td>After-Hours</td></tr><tr><td>7</td><td>Holiday</td></tr></table></td></tr>
<tr><td>625</td><td>> TradingSessionSubID</td><td>N</td><td>Trading-Session-SubID (enum)</td><td>See description for TradingSessionID (336).<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Pre-Trading</td></tr><tr><td>3</td><td>Continuous</td></tr><tr><td>4</td><td>Closing</td></tr><tr><td>5</td><td>Post-Trading</td></tr><tr><td>7</td><td>Quiescent</td></tr></table></td></tr>
<tr><td>25044</td><td>> TESTradSesStatus</td><td>N</td><td>TradSes-Status (enum)</td><td>See description for TradingSessionID (336).<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Halted</td></tr><tr><td>2</td><td>Open</td></tr><tr><td>3</td><td>Closed</td></tr><tr><td>5</td><td>Pre-Close</td></tr></table></td></tr>
</table>

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>326</td><td>> SecurityTradingStatus</td><td>N</td><td>Security-Trading-Status (enum)</td><td>See description for TradingSessionID (336).
Trading status of an instrument.
2 = Trading Halt, 212 = IPO Auction, 213 = IPO Auction Freeze applicable for cash market instruments only.
214 = Pre Call applicable for trading model Continuous Auction Issuer and Specialist only.
215 = Call applicable for trading model Continuous Auction Issuer only.
216 = Freeze applicable for trading model Continuous Auction Specialist only.
217 = Trade At Close is used for trades resulting from Trade At Close security trading phase and is applicable for cash market instruments only.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>2</td><td>Trading Halt</td></tr>
<tr><td>200</td><td>Closed</td></tr>
<tr><td>201</td><td>Restricted</td></tr>
<tr><td>202</td><td>Book</td></tr>
<tr><td>203</td><td>Continuous</td></tr>
<tr><td>204</td><td>Opening Auction</td></tr>
<tr><td>205</td><td>Opening Auction Freeze</td></tr>
<tr><td>206</td><td>Intraday Auction</td></tr>
<tr><td>207</td><td>Intraday Auction Freeze</td></tr>
<tr><td>208</td><td>Circuit Breaker Auction</td></tr>
<tr><td>209</td><td>Circuit Breaker Auction Freeze</td></tr>
<tr><td>210</td><td>Closing Auction</td></tr>
<tr><td>211</td><td>Closing Auction Freeze</td></tr>
<tr><td>212</td><td>IPO Auction</td></tr>
<tr><td>213</td><td>IPO Auction Freeze</td></tr>
<tr><td>214</td><td>Pre Call</td></tr>
<tr><td>215</td><td>Call</td></tr>
<tr><td>216</td><td>Freeze</td></tr>
<tr><td>217</td><td>Trade At Close</td></tr>
<tr><td>220</td><td>Circuit Breaker Auction Triggered By Static Limit Breach</td></tr>
<tr><td>221</td><td>Circuit Breaker Auction Triggered By Static Limit Breach Freeze</td></tr>
</table>
</td></tr>
<tr><td>2705</td><td>> MarketCondition</td><td>N</td><td>Market-Condition (enum)</td><td>Indicator for stressed market conditions.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Normal</td></tr>
<tr><td>1</td><td>Stressed</td></tr>
</table>
</td></tr>
<tr><td>2447</td><td>> FastMarketIndicator</td><td>N</td><td>Fast-Market-Indicator (enum)</td><td>See description for TradingSessionID (336).
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>No</td></tr>
<tr><td>1</td><td>Yes</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>1174</td><td>> SecurityTradingEvent</td><td>N</td><td>Security-Trading-Event (enum)</td><td>Applicable for cash market products only.<table><tr><th>Value</th><th>Description</th></tr><tr><td>10</td><td>Price volatility, auction is extended</td></tr><tr><td>11</td><td>Price volatility, auction is extended again</td></tr></table></td></tr>
<tr><td>28872</td><td>> PotentialSecurity-TradingEvent</td><td>N</td><td>Security-Trading-Event (enum)</td><td>Applicable for cash market products only. When absent, there is no potential security trading event signalled.<table><tr><th>Value</th><th>Description</th></tr><tr><td>10</td><td>Price volatility, auction is extended</td></tr></table></td></tr>
<tr><td>25155</td><td>> SoldOutIndicator</td><td>N</td><td>Sold-Out-Indicator (enum)</td><td>Applicable for cash market products only (trading model continuous auction).<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Sold-out</td></tr></table></td></tr>
<tr><td>332</td><td>> HighPx</td><td>N</td><td>decimal</td><td>Upper boundary price. See 11.3.6 Instrument state change message, pg. 141.</td></tr>
<tr><td>333</td><td>> LowPx</td><td>N</td><td>decimal</td><td>Lower boundary price. See 11.3.6 Instrument state change message, pg. 141</td></tr>
<tr><td>277</td><td>> TradeCondition</td><td>N</td><td>Trade-Condition (set)</td><td>AW = Last Auction Price cannot be combined with any other value and has its own entry in order to convey the auction type through TrdType (828).
XR = Retail price is used to report prices resulting from Retail orders (applicable for cash market products only).
BB = Midpoint Price is used to report prices resulting from midpoint orders (applicable for cash market products only).<table><tr><th>Value</th><th>Description</th></tr><tr><td>U</td><td>Exchange Last</td></tr><tr><td>R</td><td>Opening Price</td></tr><tr><td>AX</td><td>High Price</td></tr><tr><td>AY</td><td>Low Price</td></tr><tr><td>AJ</td><td>Official Closing Price</td></tr><tr><td>AW</td><td>Last Auction Price</td></tr><tr><td>BD</td><td>Previous Closing Price</td></tr><tr><td>XR</td><td>Retail</td></tr><tr><td>BB</td><td>Midpoint Price</td></tr></table></td></tr>
<tr><td>442</td><td>> MultiLegReportingType</td><td>N</td><td>MultiLeg-Reporting-Type (enum)</td><td>Only applicable for TES trades of derivatives market products.
2 = Individual Leg Of A MultiLeg Security Used to report a TES leg trade price of a complex instrument trade.
3 = Multi Leg Security Used to report a TES trade price on the complex instrument.<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Single Security</td></tr><tr><td>2</td><td>Individual Leg Of A MultiLeg Security</td></tr><tr><td>3</td><td>Multi Leg Security</td></tr></table></td></tr>
</table>

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>28750</td><td>> MultiLegPriceModel</td><td>N</td><td>MultiLeg-PriceModel (enum)</td><td>Only applicable for TES trades of derivatives market products.
1 = User Defined Used to report TES leg trade prices entered by a user.<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Standard</td></tr><tr><td>1</td><td>User Defined</td></tr></table></td></tr>
<tr><td>276</td><td>> QuoteCondition</td><td>N</td><td>Quote-Condition (enum)</td><td>F = Crossed book only present for IPS (prices without matching restrictions related to quantities that lead to a crossed book).
Z = Order imbalance together with MDEntryType 0 = Bid or 1 = Offer to define a surplus (applicable for cash market products only) (see 9.3).<table><tr><th>Value</th><th>Description</th></tr><tr><td>F</td><td>Crossed book</td></tr><tr><td>Z</td><td>Order imbalance</td></tr></table></td></tr>
<tr><td>270</td><td>> MDEntryPx</td><td>N</td><td>decimal</td><td>Price.</td></tr>
<tr><td>271</td><td>> MDEntrySize</td><td>N</td><td>decimal</td><td>Quantity or trade volume when MDEntryType is 2 = Trade or B = Trade Volume. TES disclosed quantity when MDOriginType is 1 = Off-Book.</td></tr>
<tr><td>346</td><td>> NumberOfOrders</td><td>N</td><td>uInt32</td><td></td></tr>
<tr><td>1023</td><td>> MDPriceLevel</td><td>N</td><td>uInt32</td><td>Book level. Absent for implied bid/offer prices.</td></tr>
<tr><td>273</td><td>> MDEntryTime</td><td>N</td><td>timestamp</td><td>Time of entry in nanoseconds for last trade entry (TradeCondition U = Exchange Last) and for one auction (TradeCondition AW = Last Auction Price) entry (e.g. when there are opening, intraday and volatility auction trades, only for the last auction the time is given). Statistics do not have an official timestamp in the snapshot, even if they happen to be identical to the last trade and be part of the same entry.</td></tr>
<tr><td>28873</td><td>> NonDisclosedTradeVolume</td><td>N</td><td>decimal</td><td>Contains the TES trade volume that is not displayed during the day. Only present for MDEntryType B = Trade Volume. Used when trade volume is finally disclosed and also for recovery.</td></tr>
<tr><td>6139</td><td>> TotalNumberOfTrades</td><td>N</td><td>uInt32</td><td>Total Number of trades during the day. Only present for MDEntryType = B. Applicable for cash market products only.</td></tr>
<tr><td colspan="5"><MDSshGrp> sequence ends</td></tr>
</table>

11.3.2 Depth incremental message

Delivered on: T7 EMDI incremental feed, T7 MDI data feed

This message provides order book updates and trades. Order book updates are available during Trading and Fast Trading states.

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>35</td><td>MsgType</td><td>Y</td><td>string</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>X</td><td>Market Data Incremental Refresh</td></tr></table></td></tr>
<tr><td>34</td><td>MsgSeqNum</td><td>Y</td><td>uInt32</td><td>The sequence number is incremented per product across all message types on a particular feed.</td></tr>
<tr><td>49</td><td>SenderCompID</td><td>Y</td><td>uInt32</td><td>Unique id of a sender.</td></tr>
<tr><td>1300</td><td>MarketSegmentID</td><td>Y</td><td>uInt32</td><td>Product identifier, e.g. "89".</td></tr>
<tr><td colspan="5"><MDIncGrp> sequence starts</td></tr>
<tr><td>268</td><td>NoMDEntries</td><td>Y</td><td>length</td><td></td></tr>
<tr><td>1024</td><td>> MDOriginType</td><td>Y</td><td>MDOrigin-Type (enum)</td><td>0 = Book is for on-exchange trading.
1 = Off-Book is for TES trading only. Only applicable for MDEntryType 2 = Trade or B = Trade Volume.<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Book</td></tr><tr><td>1</td><td>Off-Book</td></tr></table></td></tr>
<tr><td>279</td><td>> MDUpdateAction</td><td>Y</td><td>MDUpdate-Action (enum)</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>New</td></tr><tr><td>1</td><td>Change</td></tr><tr><td>2</td><td>Delete</td></tr><tr><td>3</td><td>Delete Thru</td></tr><tr><td>4</td><td>Delete From</td></tr><tr><td>5</td><td>Overlay</td></tr></table></td></tr>
<tr><td>269</td><td>> MDEntryType</td><td>Y</td><td>MDEntry-Type (enum)</td><td>See Depth snapshot message 11.3.1.
B = Trade Volume Trade volume entry for MDI, to provide new total trade volume from the last netting interval. Also used in EMDI for recovery purposes after a failover on the exchange side. In this case, the total traded volume of units traded during the day can be found in the MDEntrySize field. For MDOriginType 1 = Off-Book, "B" is also send together with NonDisclosedTradeVolume and MDEntrySize when TESTrad-SesStatus changes to 5 = Pre-Close.
b = Market Bid, c = Market Offer Applicable for cash market products only.<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Bid</td></tr><tr><td>1</td><td>Offer</td></tr><tr><td>2</td><td>Trade</td></tr><tr><td>Q</td><td>Auction Clearing Price</td></tr><tr><td>B</td><td>Trade Volume</td></tr><tr><td>b</td><td>Market Bid</td></tr><tr><td>c</td><td>Market Offer</td></tr><tr><td>A</td><td>Imbalance</td></tr></table></td></tr>
<tr><td>48</td><td>> SecurityID</td><td>Y</td><td>int64</td><td>Instrument identifier, e.g. "8852".</td></tr>
</table>

Tag	Field Name	Req'd	Data Type	Description
22	> SecurityIDSource	Y	string	Source Identification. **Value** / **Description** M – Marketplace-assigned Identifier
270	> MDEntryPx	N	decimal	Price of market data (trade or order).
271	> MDEntrySize	N	decimal	Quantity or trade volume when MDEntryType = 2 or "B". TES disclosed quantity when MDOriginType 1 = Off-Book
346	> NumberOfOrders	N	uInt32	
1023	> MDPriceLevel	N	uInt32	Book level. Absent for implied bid/offer prices.
273	> MDEntryTime	N	timestamp	For bids and offers the official time of book entry, for trades official time of execution (all in nanoseconds).
28872	> PotentialSecurity-TradingEvent	N	Security-Trading-Event (enum)	Applicable for cash market products only. When *absent*, there is no change in potential security trading event. **0 = None** Signals a reset. Snapshot will change to *absent* in this case. **Value** / **Description** 0 – None 10 – Price volatility, auction is extended
276	> QuoteCondition	N	Quote-Condition (enum)	See *Depth snapshot* message 11.3.1. **Value** / **Description** F – Crossed book Z – Order imbalance
<TradeEntryGrp> (optional) group starts				

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>828</td><td>> TrdType</td><td>N</td><td>TrdType (enum)</td><td>See Depth snapshot message 11.3.1.

For MDOriginType 0 = Book defines when the trade happens. Only present for MDEntryType 2 = Trade and TradeCondition AW = Last Auction Price, with the exception of cross auction trades which do not establish a last auction price, i.e. TradeCondition AW is not set for these trades.

For on-exchange trades outside the auctions, this field is not set.

1006 = Xetra / Eurex Enlight triggered Trade Indicates an Eurex Enlight triggered trade.

1007 = Block QTPIP Trade Indicates an Qualified Third Party Information Provider trade.

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>Block Trade</td></tr>
<tr><td>2</td><td>Exchange For Physical (EFP)</td></tr>
<tr><td>12</td><td>Exchange For Swap (EFS)</td></tr>
<tr><td>54</td><td>OTC</td></tr>
<tr><td>55</td><td>Exchange Basis Facility</td></tr>
<tr><td>1000</td><td>Vola Trade</td></tr>
<tr><td>1001</td><td>EFP-Fin Trade</td></tr>
<tr><td>1002</td><td>EFP-Index-Futures Trade</td></tr>
<tr><td>1004</td><td>Block Trade at Market</td></tr>
<tr><td>1006</td><td>Xetra / Eurex Enlight triggered Trade</td></tr>
<tr><td>1007</td><td>Block QTPIP Trade</td></tr>
<tr><td>1017</td><td>Delta Trade At Market</td></tr>
<tr><td>1100</td><td>Opening Auction Trade</td></tr>
<tr><td>1101</td><td>Intraday Auction Trade</td></tr>
<tr><td>1102</td><td>Volatility Auction Trade</td></tr>
<tr><td>1103</td><td>Closing Auction Trade</td></tr>
<tr><td>1104</td><td>Cross Auction Trade</td></tr>
<tr><td>1107</td><td>IPO Auction Trade</td></tr>
<tr><td>1108</td><td>Liquidity Improvement Cross</td></tr>
</table>
</td></tr>
<tr><td>2667</td><td>> AlgorithmicTrade-Indicator</td><td>N</td><td>Algorithmic-Trade-Indicator (enum)</td><td>A trade has to be flagged as "algorithmic", if at least one of the matched orders was submitted by a trading algorithm. Applicable for cash market products only.

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>Algorithmic Trade</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>277</td><td>> TradeCondition</td><td>N</td><td>Trade-Condition (set)</td><td>Defines the type of price for MDEntryPx. Only present for MDEntryType 2 = Trade. See also Depth snapshot message 11.3.1.
a = Volume Only used for coherent entries from direct matching of complex instruments (mutually exclusive with U).
k = Out of sequence is mutually exclusive with U. It is used for trades entered manually by Market Supervision, CLIP trades outside BBO which are reported as Liquidity Improvement Cross and for simple instrument Off-Book trades which are part of a basket trade.
SA = Special Auction Indicates a Special Auction Price and is only applicable for trading model Continuous Auction Specialist. For federal bonds the Special Auction indicator is used for prices determined with Bundesbank participation.
BC = Trading On Terms Of Issue Only applicable for cash market products.
TC = Trade At Close is used for trade prices resulting from Trade At Close security trading phase and is applicable for cash market instruments only.
XR = Retail price. Only applicable for cash market products.
BB = Midpoint Price. Only applicable for cash market products.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>U</td><td>Exchange Last</td></tr>
<tr><td>R</td><td>Opening Price</td></tr>
<tr><td>AX</td><td>High Price</td></tr>
<tr><td>AY</td><td>Low Price</td></tr>
<tr><td>AJ</td><td>Official Closing Price</td></tr>
<tr><td>AW</td><td>Last Auction Price</td></tr>
<tr><td>k</td><td>Out of sequence</td></tr>
<tr><td>BD</td><td>Previous Closing Price</td></tr>
<tr><td>a</td><td>Volume Only</td></tr>
<tr><td>BC</td><td>Trading On Terms Of Issue</td></tr>
<tr><td>SA</td><td>Special Auction</td></tr>
<tr><td>TC</td><td>Trade At Close</td></tr>
<tr><td>XR</td><td>Retail</td></tr>
<tr><td>BB</td><td>Midpoint Price</td></tr>
</table>
</td></tr>
<tr><td>442</td><td>> MultiLegReportingType</td><td>N</td><td>MultiLeg-Reporting-Type (enum)</td><td>See Depth snapshot message 11.3.1.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>Single Security</td></tr>
<tr><td>2</td><td>Individual Leg Of A MultiLeg Security</td></tr>
<tr><td>3</td><td>Multi Leg Security</td></tr>
</table>
</td></tr>
<tr><td>28750</td><td>> MultiLegPriceModel</td><td>N</td><td>MultiLeg-PriceModel (enum)</td><td>See Depth snapshot message 11.3.1.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Standard</td></tr>
<tr><td>1</td><td>User Defined</td></tr>
</table>
</td></tr>
<tr><td>2445</td><td>> AggressorTime</td><td>N</td><td>timestamp</td><td>Entry time of the incoming order that triggered the trade. Only present for MDEntryType=2.</td></tr>
<tr><td>483</td><td>> TransBkdTime</td><td>N</td><td>timestamp</td><td>Booked time of the off exchange trade. Can only be present for MDOriginType 1 = Off-Book.</td></tr>
</table>

Tag	Field Name	Req'd	Data Type	Description
5979	> RequestTime	N	timestamp	Gateway-In timestamp.
2446	> AggressorSide	N	Aggressor-Side (enum)	Side of the incoming order that triggered the trade. Only present for MDEntryType=2. **Value** / **Description** 1 – Buy 2 – Sell
2449	> NumberOfBuyOrders	N	uInt32	Number of buy orders involved in the trade. Only present for MDEntryType=2 and Trade Condition other than "a" (Volume Only).
2450	> NumberOfSellOrders	N	uInt32	Number of sell orders involved in the trade. Only present for MDEntryType=2 and Trade Condition other than "a" (Volume Only).
28911	> NumberOfBuySides	N	uInt32	Number of buy sides involved in an off exchange trade. Only present for MDOriginType 1 = Off-Book.
28912	> NumberOfSellSides	N	uInt32	Number of sell sides involved in an off exchange trade. Only present for MDOriginType 1 = Off-Book.
6139	> TotalNumberOfTrades	N	uInt32	Total Number of trades during the day. Only present for MDEntryType=2. Applicable for cash market products only. An increment of *TotalNumberOfTrades* is defined as the maximum of *NumberOfBuyOrders (2449)* and *NumberOfSellOrders (2450)* per trade.
28869	> RestingCxlQty	N	decimal	Quantity that was cancelled due to SMP. Only present for MDEntryType=2.
278	> MDEntryID	N	uInt32	Represents the match step ID. This field is unique together with MarketSegmentID. Only present for MDEntryType = 2.
28873	> NonDisclosedTradeVolume	N	decimal	Contains the TES trade volume that is not displayed during the day. Only present for MDEntryType B = Trade Volume. Used when trade volume is finally disclosed and also for recovery.
<TradeEntryGrp> (optional) group ends				
<MDIncGrp> sequence ends				

11.3.3 Top Of Book Implied message

Delivered on: T7 EMDI incremental feed, T7 MDI data feed

This data message is used to convey top of book information resulting from synthetic IPS matching opportunities.

Tag	Field Name	Req'd	Data Type	Description
35	MsgType	Y	string	**Value** / **Description**: X – Market Data Incremental Refresh
34	MsgSeqNum	N	uInt32	The sequence number of the message is incremented per product across different message types.
49	SenderCompID	Y	uInt32	Unique id of a sender.
1300	MarketSegmentID	Y	uInt32	Product id or pool id.
<MDIncGrp> sequence starts				
268	NoMDEntries	Y	length	
279	> MDUpdateAction	Y	MDUpdate-ActionType (enum)	**Value** / **Description**: 0 – To establish a new implied price.; 2 – To remove an existing price.
269	> MDEntryType	Y	MDEntry-Type (enum)	**Value** / **Description**: 0 – Bid; 1 – Offer
1021	> MDBookType	Y	MDBook-Type (enum)	Always Top of Book. **Value** / **Description**: 1 – Top of Book
1173	> MDSubBookType	Y	MDSub-Book-Type (enum)	Price / Quantity pairs. **Value** / **Description**: 1 – IPS implied volume without quantity restriction; 2 – IPS implied volume with quantity restriction
48	> SecurityID	Y	int64	Internal identifier assigned to each instrument.
22	> SecurityIDSource	Y	string	**Value** / **Description**: M – Marketplace-assigned identifier
270	> MDEntryPx	Y	decimal	Best implied price with or without a restriction.
271	> MDEntrySize	N	decimal	Quantity.
273	> MDEntryTime	N	timestamp	For bids and offers the official time of book entry.
276	> QuoteCondition	N	Quote-Condition (enum)	Only present for prices without matching restrictions related to quantities that lead to a crossed book if displayed. Not used for prices with quantity restrictions (see *MDSubBook-Type (1173)*. **Value** / **Description**: F – Crossed book.
<MDIncGrp> sequence ends				

11.3.4 Product state change message

Delivered on: T7 EMDI incremental feed, T7 MDI data feed

The *Product state change* message provides permanent updates on the trading state for a particular product.

Tag	Field Name	Req'd	Data Type	Description
35	MsgType	Y	string	**Value** / **Description**: h – Trading Session Status
34	MsgSeqNum	Y	uInt32	The sequence number is incremented per product across all message types on a particular feed.
49	SenderCompID	Y	uInt32	Unique id of a sender.
1300	MarketSegmentID	Y	uInt32	Product identifier, e.g. "89".
336	TradingSessionID	Y	Trading-SessionID (enum)	**Value** / **Description**: 1 – Day; 3 – Morning; 5 – Evening; 6 – After-Hours; 7 – Holiday
625	TradingSessionSubID	Y	Trading-Session-SubID (enum)	**Value** / **Description**: 1 – Pre-Trading; 3 – Continuous; 4 – Closing; 5 – Post-Trading; 7 – Quiescent
340	TradSesStatus	Y	TradSes-Status (enum)	**Value** / **Description**: 1 – Halted; 2 – Open; 3 – Closed
2705	MarketCondition	N	Market-Condition (enum)	See *Depth snapshot* message 11.3.1. **Value** / **Description**: 0 – Normal; 1 – Stressed
2447	FastMarketIndicator	Y	Fast-Market-Indicator (enum)	Indicates if product is in state "Fast Market". **Value** / **Description**: 0 – No; 1 – Yes
60	TransactTime	Y	timestamp	
25044	TESTradSesStatus	N	TradSes-Status (enum)	**Value** / **Description**: 1 – Halted; 2 – Open; 3 – Closed; 5 – Pre-Close

11.3.5 Mass instrument state change message

Delivered on: T7 EMDI incremental feed, T7 MDI data feed

The *Mass instrument state change* message provides the state information for all instruments of a certain instrument type within a product. Where not all indicated instruments are affected by the new state, the exception list (*SecurityTradingStatus (326)*) is populated with one entry for each such instrument.

Under Fast Market conditions, this message is sent with the *FastMarketIndicator (2447)* set but the actual state information may not have changed and is simply a re-statement of the previous information.

A state change affecting a single instrument (such as an intraday expiration) does not trigger a *mass instrument state change*.

Tag	Field Name	Req'd	Data Type	Description
35	MsgType	Y	string	**Value** / **Description**: CO – Security Mass Status
34	MsgSeqNum	Y	uInt32	The sequence number is incremented per product across all message types on a particular feed.
49	SenderCompID	Y	uInt32	Unique id of a sender.
1300	MarketSegmentID	Y	uInt32	Product identifier, e.g. "89".
1544	InstrumentScopeProduct-Complex	Y	Instrument-Scope-Product-Complex (enum)	Instrument type of affected instruments. **Value** / **Description**: 1 – Simple Instrument; 2 – Standard Option Strategy; 3 – Non-Standard Option Strategy; 4 – Volatility Strategy; 5 – Futures Spread; 6 – Inter-Product Spread; 7 – Standard Futures Strategy; 8 – Pack and Bundle; 9 – Strip; 10 – Flexible Instrument; 11 – Commodity Strips; 12 – Scaled Simple Instrument; 13 – Non-Standard Volatility Strategy; 14 – Total Return Future Strategy
30965	SecurityMassStatus	Y	Security-Status (enum)	The instrument status of all affected instruments. See SecurityStatus in *Depth Snapshot* 11.3.1. **Value** / **Description**: 1 – Active; 2 – Inactive; 4 – Expired; 6 – Knocked-Out; 7 – Knock-Out Revoked; 9 – Suspended; 11 – Pending Deletion; 12 – Knocked-Out And Suspended

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>1679</td><td>SecurityMassTradingStatus</td><td>N</td><td>SecurityTradingStatus (enum)</td><td>See SecurityTradingStatus in Depth Snapshot 11.3.1.
Empty for flexible instruments.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>2</td><td>Trading Halt</td></tr><tr><td>200</td><td>Closed</td></tr><tr><td>201</td><td>Restricted</td></tr><tr><td>202</td><td>Book</td></tr><tr><td>203</td><td>Continuous</td></tr><tr><td>204</td><td>Opening Auction</td></tr><tr><td>205</td><td>Opening Auction Freeze</td></tr><tr><td>206</td><td>Intraday Auction</td></tr><tr><td>207</td><td>Intraday Auction Freeze</td></tr><tr><td>208</td><td>Circuit Breaker Auction</td></tr><tr><td>209</td><td>Circuit Breaker Auction Freeze</td></tr><tr><td>210</td><td>Closing Auction</td></tr><tr><td>211</td><td>Closing Auction Freeze</td></tr><tr><td>212</td><td>IPO Auction</td></tr><tr><td>213</td><td>IPO Auction Freeze</td></tr><tr><td>214</td><td>Pre Call</td></tr><tr><td>215</td><td>Call</td></tr><tr><td>216</td><td>Freeze</td></tr><tr><td>217</td><td>Trade At Close</td></tr><tr><td>220</td><td>Circuit Breaker Auction Triggered By Static Limit Breach</td></tr><tr><td>221</td><td>Circuit Breaker Auction Triggered By Static Limit Breach Freeze</td></tr></table></td></tr>
<tr><td>28894</td><td>MassMarketCondition</td><td>Y</td><td>MarketCondition (enum)</td><td>See Depth snapshot message 11.3.1.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Normal</td></tr><tr><td>1</td><td>Stressed</td></tr></table></td></tr>
<tr><td>2447</td><td>FastMarketIndicator</td><td>Y</td><td>FastMarketIndicator (enum)</td><td>Indicates if product is in state "Fast Market". This indicator refers to a product but is provided on instrument level.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>1680</td><td>SecurityMassTradingEvent</td><td>N</td><td>SecurityTradingEvent (enum)</td><td>Identifies an event related to a SecurityMassTradingStatus(1679). Applicable for cash market products only.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>10</td><td>Price volatility, auction is extended</td></tr><tr><td>11</td><td>Price volatility, auction is extended again</td></tr></table></td></tr>
<tr><td>35155</td><td>MassSoldOutIndicator</td><td>N</td><td>SoldOutIndicator (enum)</td><td>Identifies the sold out status of all affected instruments (applicable for Continuous Auction Issuer trading model only).
See SoldOutIndicator in Depth Snapshot 11.3.1.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Sold out</td></tr></table></td></tr>
</table>

Tag	Field Name	Req'd	Data Type	Description
60	TransactTime	Y	timestamp	Time when request was processed by the matcher (nanoseconds).
35045	TESSecurityMassStatus	N	Security-Status (enum)	Status of all affected instruments for TES trading. See TESSecurityStatus in *Depth Snapshot* 11.3.1. **Value** / **Description**: 1 Active; 2 Inactive; 4 Expired; 9 Suspended
<SecMassStatGrp> sequence starts				
146	NoRelatedSym	N	length	
48	> SecurityID	Y	int64	Instrument identifier, e.g. "8852".
22	> SecurityIDSource	Y	string	**Value** / **Description**: M Marketplace-assigned Identifier
965	> SecurityStatus	Y	Security-Status (enum)	See *Depth snapshot* message 11.3.1. **Value** / **Description**: 1 Active; 2 Inactive; 4 Expired; 6 Knocked-Out; 7 Knock-Out Revoked; 9 Suspended; 11 Pending Deletion; 12 Knocked-Out And Suspended

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>326</td><td>> SecurityTradingStatus</td><td>N</td><td>Security-Trading-Status (enum)</td><td>See Depth snapshot message 11.3.1.
Empty for flexible instruments.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>2</td><td>Trading Halt</td></tr><tr><td>200</td><td>Closed</td></tr><tr><td>201</td><td>Restricted</td></tr><tr><td>202</td><td>Book</td></tr><tr><td>203</td><td>Continuous</td></tr><tr><td>204</td><td>Opening Auction</td></tr><tr><td>205</td><td>Opening Auction Freeze</td></tr><tr><td>206</td><td>Intraday Auction</td></tr><tr><td>207</td><td>Intraday Auction Freeze</td></tr><tr><td>208</td><td>Circuit Breaker Auction</td></tr><tr><td>209</td><td>Circuit Breaker Auction Freeze</td></tr><tr><td>210</td><td>Closing Auction</td></tr><tr><td>211</td><td>Closing Auction Freeze</td></tr><tr><td>212</td><td>IPO Auction</td></tr><tr><td>213</td><td>IPO Auction Freeze</td></tr><tr><td>214</td><td>Pre Call</td></tr><tr><td>215</td><td>Call</td></tr><tr><td>216</td><td>Freeze</td></tr><tr><td>217</td><td>Trade At Close</td></tr><tr><td>220</td><td>Circuit Breaker Auction Triggered By Static Limit Breach</td></tr><tr><td>221</td><td>Circuit Breaker Auction Triggered By Static Limit Breach Freeze</td></tr></table></td></tr>
<tr><td>2705</td><td>> MarketCondition</td><td>Y</td><td>Market-Condition (enum)</td><td>See Depth snapshot message 11.3.1.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Normal</td></tr><tr><td>1</td><td>Stressed</td></tr></table></td></tr>
<tr><td>1174</td><td>> SecurityTradingEvent</td><td>N</td><td>Security-Trading-Event (enum)</td><td>Identifies an event related to a SecurityTradingStatus(326). Applicable for cash market products only.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>10</td><td>Price volatility, auction is extended</td></tr><tr><td>11</td><td>Price volatility, auction is extended again</td></tr></table></td></tr>
<tr><td>25155</td><td>> SoldOutIndicator</td><td>N</td><td>Sold-Out-Indicator (enum)</td><td>Identifies the sold out status (applicable for Continuous Auction Issuer trading model only).
<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Sold out</td></tr></table></td></tr>
<tr><td>332</td><td>> HighPx</td><td>N</td><td>decimal</td><td>Upper boundary price. See 11.3.6 Instrument state change message, pg. 141.</td></tr>
<tr><td>333</td><td>> LowPx</td><td>N</td><td>decimal</td><td>Lower boundary price. See 11.3.6 Instrument state change message, pg. 141.</td></tr>
</table>

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>25045</td><td>> TESSecurityStatus</td><td>N</td><td>Security-Status (enum)</td><td>Status of the instrument for TES trading.
See TESSecurityStatus in Depth Snapshot 11.3.1.
<table><tr><th>Value</th><th>Description</th></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>
<tr><td colspan="5"><SecMassStatGrp> sequence ends</td></tr>
<tr><td>893</td><td>LastFragment</td><td>Y</td><td>Last-Fragment (enum)</td><td>Indicates whether this message is the last in a sequence of messages that together convey a joint exception list of SecMassStatGrp. All messages up to the last with LastFragment = Y share the same root level content and an application first needs to combine all single exception lists before the Mass State Change message could be applied with the fully joint exception list
<table><tr><th>Value</th><th>Description</th></tr><tr><td>N</td><td>Not Last Message</td></tr><tr><td>Y</td><td>Last Message</td></tr></table></td></tr>
</table>

11.3.6 Instrument state change message

Delivered on: T7 EMDI incremental feed, T7 MDI data feed

The *Instrument state change* message provides state information for a single instrument. It also informs participants about intraday expirations of instruments. In that case the field *SecurityStatus (965)* is set to 4 = Expired.

Tag	Field Name	Req'd	Data Type	Description
35	MsgType	Y	string	**Value** / **Description**: f – Security Status
34	MsgSeqNum	Y	uInt32	The sequence number is incremented per product across all message types on a particular feed.
49	SenderCompID	Y	uInt32	Unique id of a sender.
1300	MarketSegmentID	Y	uInt32	Product identifier, e.g. "89".
48	SecurityID	Y	int64	Instrument identifier, e.g. "8852".
22	SecurityIDSource	Y	string	**Value** / **Description**: M – Marketplace-assigned identifier
965	SecurityStatus	Y	Security-Status (enum)	See *Depth snapshot* message 11.3.1. **Value** / **Description**: 1 – Active; 2 – Inactive; 4 – Expired; 6 – Knocked-Out; 7 – Knock-Out Revoked; 9 – Suspended; 11 – Pending Deletion; 12 – Knocked-Out And Suspended

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>326</td><td>SecurityTradingStatus</td><td>N</td><td>Security-Trading-Status (enum)</td><td>See Depth snapshot message 11.3.1.
Empty for flexible instruments.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>2</td><td>Trading Halt</td></tr><tr><td>200</td><td>Closed</td></tr><tr><td>201</td><td>Restricted</td></tr><tr><td>202</td><td>Book</td></tr><tr><td>203</td><td>Continuous</td></tr><tr><td>204</td><td>Opening Auction</td></tr><tr><td>205</td><td>Opening Auction Freeze</td></tr><tr><td>206</td><td>Intraday Auction</td></tr><tr><td>207</td><td>Intraday Auction Freeze</td></tr><tr><td>208</td><td>Circuit Breaker Auction</td></tr><tr><td>209</td><td>Circuit Breaker Auction Freeze</td></tr><tr><td>210</td><td>Closing Auction</td></tr><tr><td>211</td><td>Closing Auction Freeze</td></tr><tr><td>212</td><td>IPO Auction</td></tr><tr><td>213</td><td>IPO Auction Freeze</td></tr><tr><td>214</td><td>Pre Call</td></tr><tr><td>215</td><td>Call</td></tr><tr><td>216</td><td>Freeze</td></tr><tr><td>217</td><td>Trade At Close</td></tr><tr><td>220</td><td>Circuit Breaker Auction Triggered By Static Limit Breach</td></tr><tr><td>221</td><td>Circuit Breaker Auction Triggered By Static Limit Breach Freeze</td></tr></table></td></tr>
<tr><td>2705</td><td>MarketCondition</td><td>Y</td><td>Market-Condition (enum)</td><td>See Depth snapshot message 11.3.1.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Normal</td></tr><tr><td>1</td><td>Stressed</td></tr></table></td></tr>
<tr><td>2447</td><td>FastMarketIndicator</td><td>Y</td><td>Fast-Market-Indicator (enum)</td><td>Indicates if product is in state "Fast Market". This indicator refers to a product but is provided on instrument level.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>1174</td><td>SecurityTradingEvent</td><td>N</td><td>Security-Trading-Event (enum)</td><td>Identifies an event related to a SecurityTradingStatus(326). Applicable for cash market products only.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>10</td><td>Price volatility, auction is extended</td></tr><tr><td>11</td><td>Price volatility, auction is extended again</td></tr></table></td></tr>
<tr><td>25155</td><td>SoldOutIndicator</td><td>N</td><td>Sold Out Indicator (enum)</td><td>Identifies the sold out status (applicable for Continuous Auction Issuer trading model only).
<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Sold-Out</td></tr></table></td></tr>
</table>

Tag	Field Name	Req'd	Data Type	Description
332	HighPx	N	decimal	Upper boundary price. For scheduled auctions (e.g. opening auction), it may be set together with *SecurityTradingEvent (1174)* 10 = Price volatility, auction is extended. For volatility auctions it may be set immediately without *SecurityTradingEvent (1174)*. If the value is *absent*, any previous upper boundary price is not valid any longer. Only applicable for cash market instruments.
333	LowPx	N	decimal	Lower boundary price. For scheduled auctions (e.g. opening auction), it may be set together with *SecurityTradingEvent (1174)* 10 = Price volatility, auction is extended. For volatility auctions it may be set immediately without *SecurityTradingEvent (1174)*. If the value is *absent*, any previous lower boundary price is not valid any longer. Only applicable for cash market instruments.
60	TransactTime	Y	timestamp	Time when request was processed by the matcher (nanoseconds).
25045	TESSecurityStatus	N	SecurityStatus (enum)	Status of the instrument for TES trading. See TESSecurityStatus in *Depth Snapshot* 11.3.1. **Value** / **Description** 1 – Active 2 – Inactive 4 – Expired 9 – Suspended

11.3.7 Quote request message

Delivered on: T7 EMDI incremental feed, T7 MDI data feed

Market participants can enter a quote request (Trading Interest) that needs to be answered with a quote by Market Makers. The quote request message shows such requests from traders.

Tag	Field Name	Req'd	Data Type	Description
35	MsgType	Y	string	**Value** / **Description**: R – Quote Request
34	MsgSeqNum	Y	uInt32	The sequence number is incremented per product across all message types on a particular feed.
49	SenderCompID	Y	uInt32	Unique id of a sender.
1300	MarketSegmentID	Y	uInt32	Product identifier, e.g. 89"."
<QuotReqGrp> sequence starts				
146	NoRelatedSym	Y	length	always “1”.
48	> SecurityID	Y	int64	Instrument identifier, e.g. 8852"."
22	> SecurityIDSource	Y	string	Identifies class or source of the SecurityID (48) value. Required if SecurityID is specified. Will be sent as a constant. **Value** / **Description**: M – Marketplace-assigned Identifier.
54	> Side	N	Side (enum)	**Value** / **Description**: 1 – Buy; 2 – Sell
38	> OrderQty	N	decimal	Defines the requested quantity which can be zero in a quote request.
60	> TransactTime	Y	timestamp	Time when request was processed by the matcher (nanoseconds).
<QuotReqGrp> sequence ends				

11.3.8 Cross request message

Delivered on: T7 EMDI incremental feed, T7 MDI data feed

A *crossing* is defined as intentional or unintentional execution of orders and quotes against a preselected member or in-house. Using the Cross Request, all Members are informed of a crossing or a pre-arranged trade that shall be executed in the T7 order book (on-exchange). Other market participants see the order and can also match against them. T7 expects the orders to be entered within a certain time frame. For Liquidity Improvment Crossings price, quantity and side are optional.

Tag	Field Name	Req'd	Data Type	Description
35	MsgType	Y	string	**Value** = U16 (Cross Request)
34	MsgSeqNum	Y	uInt32	The sequence number is incremented per product across all message types on a particular feed.
49	SenderCompID	Y	uInt32	Unique id of a sender.
1300	MarketSegmentID	Y	uInt32	Product identifier, e.g. 89"."
48	SecurityID	Y	int64	Instrument identifier, e.g. 8852"."
22	SecurityIDSource	Y	string	Identifies class or source of the SecurityID (48) value. Required if SecurityID is specified. Will be sent as a constant "M".
38	OrderQty	N	decimal	Defines the requested quantity which cannot be zero in a cross request.
28771	CrossRequestType	Y	Cross Request Type (enum)	**Value** = 1 (Cross Announcement); 2 (Liquidity Improvement Cross)
<CrossRequestSideGrp> sequence starts				
552	NoCrossRequestSideGrp	N	length	Always 1.
54	> Side	N	Side	**Value** = 1 (Buy); 2 (Sell)
979	> InputSource	Y	Input Source	**Value** = 1 (Client Broker)
<CrossRequestSideGrp> sequence ends				
270	Price	N	decimal	Price of the Liquidity Improvement Cross
60	TransactTime	Y	timestamp	Time when request was processed by the matcher (nanoseconds).

11.3.9 Complex instrument update message

Delivered on: T7 EMDI incremental feed, T7 MDI data feed

This message provides information for complex instruments. This message has a similar structure as the *instrument incremental message* described in section 11.2.3. Note: Complex instruments created at previous business days are published at startup. In restart / failure scenarios complex instruments are sent again. The TransactTime (60) indicates the creation / last modify time.

Tag	Field Name	Req'd	Data Type	Description
35	MsgType	Y	string	**Value** / **Description**: BP – Security Definition Update Report
34	MsgSeqNum	Y	uInt32	The sequence number is incremented per product across all message types on a particular feed.
49	SenderCompID	N	uInt32	Unique id of a sender
980	SecurityUpdateAction	Y	string	Add of complex instruments. **Value** / **Description**: A – Add
48	SecurityID	Y	int64	The instrument identifier uniquely identifies an instrument, e.g. 72057800196358145.
22	SecurityIDSource	Y	string	**Value** / **Description**: M – Marketplace-assigned identifier
107	SecurityDesc	Y	string	See *Instrument snapshot* message 11.2.2.
167	SecurityType	Y	Security-Type (enum)	Type of security. **Value** / **Description**: MLEG – complex instrument
762	SecuritySubType	N	uInt32	Standard strategy type for complex instruments. The mapping of integer values to the corresponding strategy types is available at: www.eurex.com > Data > Trading files > Product parameter files > Trading parameters
1227	ProductComplex	Y	Product-Complex (enum)	Type of instrument. **Value** / **Description**: 1 – Simple Instrument; 2 – Standard Option Strategy; 3 – Non-Standard Option Strategy; 4 – Volatility Strategy; 5 – Futures Spread; 6 – Inter-Product Spread; 7 – Standard Futures Strategy; 8 – Pack and Bundle; 9 – Strip; 10 – Flexible Instrument; 11 – Commodity Strips; 13 – Non-Standard Volatility Strategy; 14 – Total Return Future Strategy
28900	LegRatioMultiplier	N	uInt32	Common integer multiple of the option legs for Option Volatility Strategies.

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>28907</td><td>QuantityScalingFactor</td><td>N</td><td>uInt32</td><td>An integer value that determines the lot size of orders, quotes and trades. Only valid for derivatives markets products that support scaled simple instruments, see QuantityScalingFactors (11.2.1, pg. 86)</td></tr>
<tr><td>893</td><td>LastFragment</td><td>Y</td><td>LastFragment (enum)</td><td>Indicates whether this message is the last in a sequence of messages that together convey a joint list of InstrmtLegGrp. All messages up to the last with LastFragment = Y share the same root level content and an application first needs to combine all InstrmtLeg lists before it has the complete definition of an complex instrument. (Note, last fragment = N is not expected for EUREX)
<table><tr><th>Value</th><th>Description</th></tr><tr><td>N</td><td>Not Last Message</td></tr><tr><td>Y</td><td>Last Message</td></tr></table></td></tr>
<tr><td colspan="5"><InstrmtLegGrp> sequence starts</td></tr>
<tr><td>555</td><td>NoLegs</td><td>N</td><td>length</td><td></td></tr>
<tr><td>600</td><td>> LegSymbol</td><td>Y</td><td>uInt32</td><td>Product identifier of the leg security.</td></tr>
<tr><td>602</td><td>> LegSecurityID</td><td>Y</td><td>int64</td><td>Instrument identifier of the leg security.</td></tr>
<tr><td>603</td><td>> LegSecurityIDSource</td><td>Y</td><td>string</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>M</td><td>Marketplace-assigned identifier</td></tr></table></td></tr>
<tr><td>609</td><td>> LegSecurityType</td><td>Y</td><td>LegSecurityType (enum)</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>MLEG</td><td>Part of a multi-leg instrument</td></tr><tr><td>ULEG</td><td>Underlying leg</td></tr></table></td></tr>
<tr><td>624</td><td>> LegSide</td><td>Y</td><td>LegSide (enum)</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Buy</td></tr><tr><td>2</td><td>Sell</td></tr></table></td></tr>
<tr><td>623</td><td>> LegRatioQty</td><td>Y</td><td>uInt32</td><td>The ratio of quantity for this individual leg relative to the entire multileg security.</td></tr>
<tr><td>566</td><td>> LegPrice</td><td>N</td><td>decimal</td><td>Price for the leg instrument. Used for the underlying leg of a Volatility or Non-Standard Volatility Strategy.</td></tr>
<tr><td>30810</td><td>> RelatedPrice</td><td>N</td><td>decimal</td><td>This field is used to convey the price further specified by RelatedPriceType (30419).</td></tr>
<tr><td>30419</td><td>> RelatedPriceType</td><td>N</td><td>RelatedPriceType (enum)</td><td>Indicates what kind of price is reported in the field RelatedPrice (30810).
1 = Basis Refers to a Basis price in RelatedPrice (30810)
<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Basis</td></tr></table></td></tr>
</table>

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>31194</td><td>> TradingStyle</td><td>N</td><td>Trading-Style (enum)</td><td>This field is used for all legs of a Total Return Future Strategy.
1 = Regular A Regular future instrument.
2 = TRF_TAC A Total Return Futures instrument that is traded at underlying close (TAC).
3 = TRF_TAM A Total Return Futures instrument that is traded at market (TAM).
4 = MOC A MOC (Market-On-Close) futures instrument.<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Regular</td></tr><tr><td>2</td><td>TRF_TAC</td></tr><tr><td>3</td><td>TRF_TAM</td></tr><tr><td>4</td><td>MOC</td></tr></table></td></tr>
<tr><td colspan="5"><InstrmtLegGrp> sequence ends</td></tr>
<tr><td colspan="5"><MarketSegmentGrp> sequence starts</td></tr>
<tr><td>1310</td><td>> NoMarketSegments</td><td>N</td><td>length</td><td>always “1”.</td></tr>
<tr><td>1300</td><td>> MarketSegmentID</td><td>Y</td><td>uInt32</td><td>Product identifier, e.g. 89.</td></tr>
<tr><td>1144</td><td>> ImpliedMarketIndicator</td><td>N</td><td>Implied-Market-Indicator (enum)</td><td>Matching method.<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Not implied</td></tr><tr><td>3</td><td>Both implied-in and implied-out</td></tr></table></td></tr>
<tr><td colspan="5"><MarketSegmentGrp> sequence ends</td></tr>
<tr><td>60</td><td>TransactTime</td><td>N</td><td>timestamp</td><td>Creation time of complex instruments.</td></tr>
</table>

11.3.10 Flexible instrument update message

Delivered on: T7 EMDI incremental feed, T7 MDI data feed

This message provides information for new flexible instruments.

Tag	Field Name	Req'd	Data Type	Description
35	MsgType	Y	string	**Value** / **Description**: BP – Security Definition Update Report
34	MsgSeqNum	Y	uInt32	The sequence number is incremented per product across all message types on a particular feed.
49	SenderCompID	Y	uInt32	Unique id of a sender
980	SecurityUpdateAction	Y	string	Intraday creation of flexible instruments. **Value** / **Description**: A – Add
48	SecurityID	Y	int64	The instrument identifier uniquely identifies a flexible instrument.
22	SecurityIDSource	Y	string	**Value** / **Description**: M – Marketplace-assigned identifier
107	SecurityDesc	Y	string	The format depends on the type of instrument. Options: OGBM FI 20140516 CS AM P 101.82 0 Futures: FGBM FI 20140620 CS
167	SecurityType	Y	Security-Type (enum)	Type of security. **Value** / **Description**: OPT – Option; FUT – Future
1227	ProductComplex	Y	Product-Complex (enum)	Type of instrument. **Value** / **Description**: 10 – Flexible Instrument
541	MaturityDate	Y	uInt32	Expiration day of the instrument (YYYYMMDD).
30866	ContractDate	Y	uInt32	Date used to identify the instrument (YYYYMMDD).
2962	SecurityReferenceData-Supplement	Y	uInt32	Date used to identify the instrument (YYYYMMDD). Same as user defined field *ContractDate (30866)*.
202	StrikePrice	N	decimal	Strike price.
201	PutOrCall	N	PutOrCall (enum)	Defines if instrument is a put or call. **Value** / **Description**: 0 – Put; 1 – Call
206	OptAttribute	N	uInt32	Version of an option.
1194	ExerciseStyle	N	Exercise-Style (enum)	Style family of an option. **Value** / **Description**: 0 – European; 1 – American

Tag	Field Name	Req'd	Data Type	Description
1193	SettlMethod	Y	SettlMethod (enum)	Settlement type. **Value** / **Description** C – Cash P – Physical
<MarketSegmentGrp> sequence starts				
1310	NoMarketSegments	Y	length	always "1".
1300	> MarketSegmentID	Y	uInt32	Product identifier or product pool identifier, e.g. 89.
<MarketSegmentGrp> sequence ends				
60	TransactTime	N	timestamp	Creation time of flexible instrument.

11.3.11 Scaled simple instrument update message

Delivered on: T7 EMDI incremental feed, T7 MDI data feed only if the owning product supports scaled simple instruments, see *QuantityScalingFactors* (11.2.1, pg. 86).

This message provides information for new scaled simple instruments.

Tag	Field Name	Req'd	Data Type	Description
35	MsgType	Y	string	**Value** / **Description**: BP – Security Definition Update Report
34	MsgSeqNum	Y	uInt32	The sequence number is incremented per product across all message types on a particular feed.
49	SenderCompID	Y	uInt32	Unique id of a sender
980	SecurityUpdateAction	Y	string	Intraday creation of scaled simple instruments. **Value** / **Description**: A – Add
48	SecurityID	Y	int64	The instrument identifier uniquely identifies a scaled simple instrument.
22	SecurityIDSource	Y	string	**Value** / **Description**: M – Marketplace-assigned identifier
107	SecurityDesc	Y	string	see *SecurityDesc (107)* (11.2.2, pg. 96)
167	SecurityType	Y	Security-Type (enum)	Type of security. **Value** / **Description**: OPT – Option; FUT – Future
1227	ProductComplex	Y	Product-Complex (enum)	Type of instrument. **Value** / **Description**: 12 – Scaled Simple Instrument
28907	QuantityScalingFactor	Y	uInt32	An integer value that determines the lot size of orders, quotes and trades.
<RelatedInstrumentGrp> group starts				
1650	RelatedSecurityID	Y	int64	The instrument identifier uniquely identifies the related simple instrument.
1651	RelatedSecurityIDSource	Y	string	**Value** / **Description**: M – Marketplace-assigned identifier
<RelatedInstrumentGrp> group end				
<MarketSegmentGrp> sequence starts				
1310	NoMarketSegments	Y	length	always "1".
1300	> MarketSegmentID	Y	uInt32	Product identifier or product pool identifier, e.g. 89.
<MarketSegmentGrp> sequence ends				
60	TransactTime	N	timestamp	Creation time of scaled simple instrument.

11.4 Data files

11.4.1 Reference data from file (T7 RDF)

As an alternative to the reference data feed, participants can receive reference data in file format. This provision is made especially for bandwidth conscious users who want to prevent an overload of their line capacity caused by joining the reference data feed.

For details regarding FIXML Schema design rules please refer to FIXimate.

T7 provides a customized FIXML schema via a set of xsd files. They can be found in zipped format at this address:

www.eurex.com > Support > Initiatives & Releases > T7 Release 13.1 > System Documentation > Market & Reference Data Interfaces

or

www.xetra.com > Technology > T7 trading architecture > System documentation > Release 13.1 > Market and Reference Data Interfaces.

As per the FIXML specification, T7 RDF field names are different from the T7 RDI field names. A mapping table between the fields can be found in section 14.3, FIXML mapping table.

The reference data files can be received via the Common Report Engine as described in the "Common Report Engine User Guide".

Similar to the T7 RDI, the T7 RDF mechanism operates on a snapshot plus incremental basis.

The initial file (“snapshot”) is created at the start of a new business day and contains all of the simple instruments (variance futures states, total return futures states and trade at reference price states) defined on the exchange at the beginning of that business day.

An incremental file is created up to every five minutes thereafter and contains the creation events for complex and flexible instruments along with the variance futures, total return futures and trade at reference price status messages disseminated in that period. Each of these incremental files must be applied to the initial file (the “snapshot”).

If there have been no changes in a given five minute interval, no file is created (i.e. empty files are not created).

Other options to receive intra day created complex instruments are described in section 9.2.7, Use case 5: Chronological order of messages for complex instrument creation.

11.4.2 File name format of the reference data files

The Reference Data files are provided in FIXML format. Product reference data and instrument reference data are provided in one file. The file format of the Start-Of-Day and intraday files looks as follows:

Content	File Name
Start-Of-Day file	[@@][report name][member][business date][mic][file set identifier][nnn].XML
Intraday updates	[@@][report name][member][business date][mic][file set identifier][nnn].XML

Table 53: *Format of the reference data files*

with:
@@ (length 2): environment number, i.e. ’90’ for production and 95 for simulation
report name (length 8): always ’FILRDF01’
member id (length 5): always ’PUBLI’
business date (length 8): format ’YYYYMMDD’

mic (length 4): MIC code, XEUR for T7 XEEE for EEX
file set identifier (length 5): remains constant for all files belonging to the same set
nnn (length 3): sequence number 000-999

Example for reference data files in simulation:
Start-Of-Day file for T7: 95FILRDF01PUBLI20120815XEURCZA80000.XML
First intraday updated file for T7: 95FILRDF01PUBLI20120815XEURCZA80001.XML
Second intraday updated file for T7: 95FILRDF01PUBLI20120815XEURCZA80002.XML
...

Note: Each market has its own file, therefore the file name contains only one MIC code.

Note: For XFRA environment two additional Reports will be provided.

File names:

FILRDF03 for Börse Frankfurt (BF) scope.

FILRDF04 for Börse Frankfurt Zertifikate (BFZ) scope.

11.4.3 Reference data file on the next business day

Complex instruments which still exist on the next business day and which have been sent as intraday file updates on the previous day are incorporated into the Start-Of-Day file on the next day.

11.4.4 Reference data file after a failover or restart of T7 RDI

The file set identifier changes after a restart or failover of the T7 RDI. The file set identifier always lexically increments within a day, allowing for easy identification of the most recently created set. During a failover or restart, the application must detect the creation of a new file set and rebuild its reference data based on the new file set.

Example:

Start-Of-Day file **after** failover : 95FILRDF01PUBLI20120815XEUR**CZLC0**000.XML
First intraday updated file **after** failover : 95FILRDF01PUBLI20120815XEUR**CZLC0**001.XML
Second intraday updated file **after** failover: 95FILRDF01PUBLI20120815XEUR**CZLC0**002.XML
...

If T7 RDI fails over, a new initial reference file is generated with a new file set identifier. This file contains all incremental messages disseminated during the day, i.e. the entire history.

If T7 RDI needs to be restarted by the exchange, a new initial reference file is generated with a new file set identifier. This file contains the currently existing complex instruments but not the entire history of incremental messages.

11.4.5 What receiving applications need to do

Applications using a sftp process need to process the following loop:

1. Find the newest Start-Of-Day file.
2. Apply all updates newer than the Start-Of-Day file.
3. Detect the creation of a new file set.

12 Multicast addresses

The reference information provided by T7 RDI contains the respective multicast channel information (i.e. multicast addresses and port numbers) for all available products and all provided market data interfaces.

For a full list of multicast addresses for T7 RDI reference data snapshot and incremental channels of all markets supported by the T7 trading architecture, please refer to the document Exchange and Settlement Network Access available at

www.eurex.com > Support > Initiatives & Releases > T7 Release 13.1 > System Documentation > Network Access

or

www.xetra.com > Technology > T7 trading architecture > System documentation > Release 13.1 > Network Access.

12.1 Reference data for T7 Enhanced Order Book Interface

The reference data information such as multicast addresses and port numbers of corresponding products, which are needed to receive public market data via T7 Enhanced Order Book Interface is available via the T7 Reference Data Interface (T7 RDI) and the T7 Reference Data Files (T7 RDF).

The *Product Snapshot* message contains the following information for products configured for T7 Enhanced Order Book Interface:

- Book Type, *MDBookType (1021)*, has value 3 (Order Depth).
- Feed Type, *MDFeedType (1022)*, has value HI (HighIncremental) for Order-By-Order incremental messages and HS (HighSnapshot) for Order-By-Order snapshot messages.
- The IP multicast address and port number of the primary T7 Enhanced Order Book feed can be obtained via fields *PrimaryServiceLocationID (2567)* and *PrimaryServiceLocationSubID (28591)*.
- The IP multicast address and port number of the secondary T7 Enhanced Order Book feed can be obtained via fields *SecondaryServiceLocationID (2568)* and *SecondaryServiceLocationSubID (28593)*.

The same information is also available via the T7 Reference Data Files (T7 RDF).

13 FAST templates

Two versions for FAST templates are offered:

- FAST templates based on version 1.2
- FAST templates compatible with version 1.1

Participants can either use a decoder which has the new FAST 1.2 feature implemented or use their existing decoder based on FAST 1.1.

The FAST XML files are provided by T7 in separate files:

- EMDIFastTemplates*.xml for all messages on the EMDI snapshot and incremental feeds.
- MDIFastTemplates*.xml for all messages on the MDI feed.
- RDIFastTemplates*.xml for all messages on the RDI snapshot and incremental feeds.

The FAST templates can be downloaded from the T7 website at:

www.eurex.com > Support > Initiatives & Releases > T7 Release 13.1 > System Documentation > Market & Reference Data Interfaces

or

www.xetra.com > Technology > T7 trading architecture > System documentation > Release 13.1 > Market and Reference Data Interfaces.

14 Appendix

14.1 Example for a XML FAST template

This example refers to chapter 5.3, Decoding the FAST-message.

```
- <template id="86" name="DepthIncremental">
- <string name="MsgType" id="35">
  <constant value="X" />
    </string>
- <uInt32 name="MsgSeqNum" id="34">
  <increment />
    </uInt32>
- <uInt32 name="SenderCompID" id="49">
  <copy />
    </uInt32>
- <uInt32 name="MarketSegmentID" id="1300">
  <copy />
    </uInt32>
- <sequence name="MDEntriesGrp">
  <length name="NoMDEntries" id="268" />
- <field name="MDUpdateAction" id="279">
  <type name="MDUpdateAction" />
    </field>
- <field name="MDEntryType" id="269">
  <type name="MDEntryType" />
    </field>
  <uInt64 name="SecurityID" id="48" />
- <string name="SecurityIDSource" id="22">
  <constant value="M" />
    </string>
- <decimal name="MDEntryPx" id="270" presence="optional">
  <delta />
    </decimal>
  <uInt32 name="MDEntrySize" id="271" presence="optional" />
  <uInt32 name="NumberOfOrders" id="346" presence="optional" />
- <uInt32 name="MDPriceLevel" id="1023" presence="optional">
  <delta />
    </uInt32>
- <timestamp name="MDEntryTime" unit="nanosecond" id="273">
  <copy />
    </timestamp>
- <group name="TradeEntryGrp" presence="optional">
  <uInt32 name="TradeVolume" id="1020" presence="optional" />
- <field name="TradeCondition" id="277" presence="optional">
  <type name="TradeConditionSet" />
    </field>
  <uInt32 name="GapIndicator" id="8719" presence="optional" />
  <timestamp name="AggressorTimestamp" unit="nanosecond" id="8720" presence="optional" />
<field name="AggressorSide" id="5797" presence="optional">
  <type name="Side" />
    </field>
  <uInt32 name="NumberOfBuyOrders" id="8721" presence="optional" />
  <uInt32 name="NumberOfSellOrders" id="8722" presence="optional" />
  <uInt32 name="MDEntryID" id="278" />
    </group>
    </sequence>
    </template>
```

Figure 19: *Example for a FAST template with repeating group*

14.2 Example for determination of the price source

14.2.1 Fully implied

Example for 9.3.1, Determination of the price sources

Precondition: Empty order book.

Action: Buy FDAX Mar 10@100, Sell FDAX Jun 10@98

Result: Fully implied price in future spread, Buy FDAX Mar/Jun 10@2 gets created (implied price is sent). Rule 1 in section 9.3.1, Determination of the price sources applies.

Tag number	Tag name	Value	Description
35	MsgType	X	MarketDataIncrementalRefresh
34	MsgSeqNum	1000	
49	SenderCompID	75	Unique id of a sender
1300	MarketSegmentID	89	Product
268	NoMDEntries	1	
279	> MDUpdateAction	0	New
269	> MDEntryType	0	Bid
48	> SecurityID	8875	Instrument
22	> SecurityIDSource	M	Marketplace-assigned identifier
270	> MDEntryPx	2	Price
271	> MDEntrySize	10	Quantity
346	> NumberOfOrders	N/A	Number of order/quotes on this level
1023	> MDPriceLevel	N/A	empty indicates implied price
273	> MDEntryTime	t_0	official time of book entry

Table 54: *Fully implied*

14.2.2 Fully outright on level 1

Example for 9.3.1, Determination of the price sources

Precondition: Fully implied price in future spread created above.

Action: Buy FDAX Mar/Jun 8@1.5

Result: Fully outright price gets created on price level 1 (Implied price is absent). No update for implied is sent as it has already been conveyed (see section 14.2.1). Rule 1 in section 9.3.1, Determination of the price sources applies.

Tag number	Tag name	Value	Description
35	MsgType	X	MarketDataIncrementalRefresh
34	MsgSeqNum	1001	
49	SenderCompID	75	Unique id of a sender
1300	MarketSegmentID	89	Product
268	NoMDEntries	1	
279	> MDUpdateAction	0	New
269	> MDEntryType	0	Bid
48	> SecurityID	8875	Instrument
22	> SecurityIDSource	M	Marketplace-assigned identifier
270	> MDEntryPx	1.5	Price
271	> MDEntrySize	8	Quantity
346	> NumberOfOrders	1	Number of order/quotes on this level
1023	> MDPriceLevel	1	Price level 1
273	> MDEntryTime	t_1	official time of book entry

Table 55: *Fully outright sent for level 1*

14.2.3 Partially implied

Example for 9.3.1, Determination of the price sources

Precondition: Fully implied price in future spread created before (see section 14.2.1):

- fully implied Buy FDAX (Mar/Jun) 10@2
- outright 8@1.5 (see section 14.2.2)

Action: Buy FDAX Mar/Jun 5@2 (new outright price in future spread)

Result: The initially fully implied price becomes partially implied. Rule 2 in section 9.3.1, Determination of the price sources) applies. The implied price (without price level) is not sent again as it has already been conveyed (see section 14.2.1).

Tag number	Tag name	Value	Description
35	MsgType	X	MarketDataIncrementalRefresh
34	MsgSeqNum	1002	
49	SenderCompID	75	Unique id of a sender
1300	MarketSegmentID	89	Product
268	NoMDEntries	1	
279	> MDUpdateAction	0	New
269	> MDEntryType	0	Bid
48	> SecurityID	8875	Instrument
22	> SecurityIDSource	M	Marketplace-assigned identifier
270	> MDEntryPx	2	Price
271	> MDEntrySize	5	Quantity
346	> NumberOfOrders	1	Number of order/quotes on this level
1023	> MDPriceLevel	1	Book level
273	> MDEntryTime	t_2	official time of book entry

Table 56: *Partially implied*

14.2.4 Several fully implied orders at Best Market

Eample for 9.3, General order book rules and mechanics

Precondition: Fully implied Buy FDAX Sep 100@117

Action: Buy FDAX Jun 200@118, Sell FDAX Jun/Sep 200@1.
This creates an implied for Buy FDAX Sep at 200@(118-1) = 200@117.

Result: The reported quantity is the largest of the quantities from the implied orders at the same price, i.e. 200@117 is reported.

Rule 1 in section 9.3.1, Determination of the price sources applies.

Tag number	Tag name	Value	Description
35	MsgType	X	MarketDataIncrementalRefresh
34	MsgSeqNum		
49	SenderCompID	75	Unique id of a sender
1300	MarketSegmentID	89	Product
268	NoMDEntries	1	
279	> MDUpdateAction	0	New
269	> MDEntryType	0	Bid
48	> SecurityID	7510	Instrument
22	> SecurityIDSource	M	Marketplace-assigned identifier
270	> MDEntryPx	117	Price
271	> MDEntrySize	200	Quantity
346	> NumberOfOrders	N/A	Number of order/quotes on this level
1023	> MDPriceLevel	N/A	empty indicates implied price
273	> MDEntryTime	t_3	official time of book entry

Table 57: *Quantities are added for two or more fully implied orders at the same price.*

14.3 FIXML mapping table

The following table provides a mapping between FIXML attributes and FIX fields. The mapping table belongs to chapter 11.4.1, Reference data from file (T7 RDF). *SecDef* refers to the *Instrument snapshot* message, *SecDefUpd* refers to the *Instrument incremental* message, *MktDef* refers to the *Product snapshot* message and *SecStat* refers to the *Variance Futures Status*, the *Total Return Futures Status* and the *Trade At Reference Price Status* messages.

The initial file ("snapshot") includes all four message types while the incremental file contains only the *SecDefUpd* and *SecStat* messages.

FIXML attributes of the T7 RDF occur in the same sequence as FIX fields of the T7 RDI.

The *Market Data Report* message is not used in the T7 RDF because the start and end of the files coincides with the MDReportEvent values.

Example:
MDReportEvent = 1 = StartOfReferenceData = Begin of file
MDReportEvent = 2 = EndOfReferenceData = End of file

FIXML Attribute	FIX Field Name	FIX Tag / Group	Repeating	FIX Messages
AcrdCoItnPrm	AccruedCollectionParameter	28885		SecStat
AcrdPmtPrm	AccruedPaymentParameter	28884		SecStat
AID	SecurityAlt	Group	Y	SecDef, SecDefUpd
AltID	SecurityAltID	455		SecDef, SecDefUpd
AltIDSrc	SecurityAltIDSource	456		SecDef, SecDefUpd
AnnlClndrDays	AnnualCalendarDays	28881		SecStat
AnnlTrdgBizDays	AnnualTradingBusinessDays	2584		SecStat
AOProdStrip	AllowOneProductStrip	28901		MktDef
ARMVM	ARMVM	2591		SecStat
AssetTyp	AssetType	1940		SecDef, SecDefUpd
AssetSubTyp	AssetSubType	29831		SecDef, SecDefUpd
AuctTyp	AuctionType	1803		MktDef
AuctTypRule	AuctionTypeRules	Group	Y	MktDef
BizDayTyp	BusinessDayType	2581		SecStat
CalcMeth	CalculationMethod	2592		SecStat
Ccy	Currency	15		MktDef, SecDef
CFI	CFICode	461		SecDef, SecDefUpd
ChkMktOrd	CheckMarketOrder	25217		MktDef
ClrPxOfst	ClearingPriceOffset	2582		SecStat
ClrPxPrm	ClearingPriceParameters	Group	Y	SecStat
ClsdBkInd	ClosedBookIndicator	28874		MktDef
CntrctCycTyp	ContractCycleType	30865		SecDef
CntrctCycSubTyp	ContractCycleSubType	31865		SecDef
CntrctDispInstr	ContractDisplayInstruction	25186		SecDef
CntrctDt	ContractDate	30866		SecDef, SecDefUpd

FIXML Attribute	FIX Field Name	FIX Tag / Group	Repeating	FIX Messages
CntrctDtTyp	ContractDateType	32865		SecDef, SecDefUpd
CntrctFreq	ContractFrequency	30867		SecDef, SecDefUpd
CntrctGenNum	ContractGenerationNumber	25034		SecDef, SecDefUpd
CntrctIDElig	ContractIdentificationEligibility	25215		SecDef, SecDefUpd
CntrctMTyp	ContractMonthType	33865		SecDef
CntrctMY	ContractMonthYear	32340		SecDef
CnvrsnMode	ConversionMode	28909		SecStat
CnvrsnModeQual	ConversionModeQualifier	28910		SecStat
CpctyGrpID	CapacityGroupID	25246		MktDef
CpnDayCnt	AccruedInterestCalculationMethod	1950		SecDef
CpnPmt	CouponPaymentDate	224		SecDef
CpnRt	CouponRate	223		SecDef
CurColtnPrm	CurrentCollectionParameter	28885		SecStat
CurPmtPrm	CurrentPaymentParameter	28883		SecStat
DecaySplit	DecaySplit	25144		MktDef
Desc	SecurityDesc	107		SecDef, SecDefUpd
DispDay	DisplayDay	25210		SecDef
DispReltvDay	DisplayRelativeDay	25220		SecDef
DispM	DisplayMonth	25211		SecDef
DispQ	DisplayQuarter	25189		SecDef
DispS	DisplaySeason	25214		SecDef
DispWk	DisplayWeek	25212		SecDef
DispDayOfWk	DisplayDayOfWeek	25239		SecDef
DispY	DisplayYear	25213		SecDef
DispName	DisplayName	28791		SecDef
DpstTyp	DepositType	28890		SecDef, SecDefUpd
DisablOnBkTrdg	DisableOnBookTrading	28800		MktDef
Dt	EventDate	866		SecDef, SecDefUpd
EfctvBizDt	EffectiveBusinessDate	2400		MktDef
EndPxRng	EndPriceRange	2552		MktDef
EndTickPxRng	EndTickPriceRange	1207		MktDef
EventTyp	EventType	865		SecDef, SecDefUpd
Evnt	Events	Group	Y	SecDef, SecDefUpd
Exch	SecurityExchange	207		SecDef, SecDefUpd
Exch	UnderlyingSecurityExchange	30308		MktDef
ExerStyle	ExerciseStyle	1194		SecDef, SecDefUpd
FastMktInd	FastMarketIndicator	2447		MktDef
FastMktPctg	FastMarketPercentage	2557		MktDef
FlatIndicator	FlatIndicator	25170		SecDef

FIXML Attribute	FIX Field Name	FIX Tag / Group	Repeating	FIX Messages
FlexProdElig	FlexProductEligibilityIndicator	1242		MktDef
FlexProdEligCmplx	FlexProductEligibilityComplex	2561		MktDef
FlexProdEligs	FlexRules	Group	Y	MktDef
HHIIntvls	HHIIntervals	Group	Y	MktDef
HHIInd	HHIIndicator	25228		MktDef
HHIIntvlEnd	HHIIntervalEnd	25229		MktDef
ID	InstrumentPartyID	1019		SecDef, SecDefUpd
ID	LegSecurityID	602		SecDef, SecDefUpd
ID	RelatedSecurityID	1650		SecDef, SecDefUpd
ID	RelatedMarketSegmentID	2546		MktDef
ID	SecurityID	48		SecDef, SecDefUpd, SecStat
ID	UnderlyingSecurityID	30309		MktDef
ID	UnderlyingSecurityID	309		SecDef
ImpldMktInd	ImpliedMarketIndicator	1144		SecDef, SecDefUpd
InpdSysOper	IndependentSystemOperator	28903		MktDef
InstrAttrib	InstrumentAttributes	Group	Y	SecDef, SecDefUpd
Instrmt	Instrument	Group	N	SecDef, SecDefUpd
InstrmtExt	InstrumentExtension	Group	N	SecDef, SecDefUpd
InstrmtScope	InstrumentScopes	Group	Y	MktDef
Issued	IssueDate	225		SecDef
IssuCtry	CountryOfIssue	470		SecDef
LctnID	LocationID	283		MktDef
Leg	InstrmtLegGrp	Group	Y	SecDef, SecDefUpd
ListMeth	ListMethod	1198		SecDef, SecDefUpd
LowExerPxOptnInd	LowExercisePriceOptionIndicator	2574		SecDef,SecDefUpd
MarketSegmentDesc	MarketSegmentDesc	1396		MktDef
MaxOfstRFQExpireTm	MaxOffsetRFQExpireTime	30126		MktDef
MidPntExecVenuID	MidpointExecVenueID	28914		SecDef, SecDefUpd
MidPntTrdgTyp	MidpointTrading	28913		SecDef, SecDefUpd
MatDt	MaturityDate	541		SecDef, SecDefUpd
MatFreqUnit	MaturityFrequencyUnit	2982		SecDef, SecDefUpd
MDBkTyp	MDBookType	1021		MktDef
MDFeedTyp	MDFeedType	1022		MktDef
MDFeedTyps	Feeds	Group	Y	MktDef
MDRcvryTmIntvl	MDRecoveryTimeInterval	2565		MktDef
MinBidSz	MinBidSize	647		MktDef, SecDef, SecDefUpd
MinOfrSz	MinOfferSize	648		MktDef, SecDef, SecDefUpd
MinPxIncr	MinPriceIncrement	969		SecDef, SecDefUpd
MinPxIncrClr	MinPriceIncrementClearing	28888		SecDef, SecDefUpd

FIXML Attribute	FIX Field Name	FIX Tag / Group	Repeating	FIX Messages
MinPxIncrAmt	MinPriceIncrementAmount	1146		SecDef, SecDefUpd
MktDepth	MarketDepth	264		MktDef
MktDepthTmIntvl	MarketDepthTimeInterval	2563		MktDef
MktID	MarketID	1301		MktDef
MktImbalInd	MarketImbalanceIndicator	28875		MktDef
MktSeg	MarketSegment	7703		MktDef
MktSegGrp	MarketSegmentGrp	Group	Y	SecDef, SecDefUpd
MktSegID	MarketSegmentID	1300		MktDef, SecDef, SecDefUpd, SecStat
MktSegID	UnderlyingMarketSegmentID	31300		SecDef
MktSegStat	MarketSegmentStatus	2542		MktDef
MktSegSubTyp	MarketSegmentSubType	2544		MktDef
MktSegTyp	MarketSegmentType	2543		MktDef
MlegModel	MultilegModel	1377		SecDef, SecDefUpd
MMY	MaturityMonthYear	200		SecDef, SecDefUpd
MtchAlgo	MatchAlgorithm	1142		MktDef
MtchRules	MatchRules	Group	Y	MktDef
MtchRuleProdCmplx	MatchRuleProductComplex	2569		MktDef
MtchTyp	MatchType	574		MktDef
Mult	ContractMultiplier	231		SecDef, SecDefUpd
NegDur	NegotiationDuration	31629		MktDef
NxtEfctvBizDt	NextEffectiveBusinessDate	28871		MktDef
OfstSTPEfctvTm	OffsetSTPEffectiveTime	30168		MktDef
Oper	InstrumentScopeOperator	1535		MktDef
OptAt	OptAttribute	206		SecDef, SecDefUpd
OrigStrkPx	OrigStrikePrice	2578		SecDef, SecDefUpd
OvrNiteIntRt	OvernightInterestRate	2590		SecStat
ParentMktSegmID	ParentMktSegmID	1325		MktDef
PartID	PartitionID	5948		MktDef
PostTrdAnon	PostTradeAnonymity	28876		SecDef, SecDefUpd
PrevClsPx	UnderlyingPrevClosePx	28843		MktDef
PrevCpnPmt	PreviousCouponPaymentDate	28895		SecDef
Pri	IsPrimary	25216		SecDef, SecDefUpd
PriSetPx	PriorSettlPrice	734		SecDef, SecDefUpd, SecStat
ProdCmplx	ProductComplex	1227		SecDef, SecDefUpd
ProdTyp	ProductType	6958		MktDef
Pty	InstrumentParties	Group	Y	SecDef, SecDefUpd
PutCall	PutOrCall	201		SecDef, SecDefUpd
CmdtyProdClss	CommodityProductClass	28902		MktDef

FIXML Attribute	FIX Field Name	FIX Tag / Group	Repeating	FIX Messages
CmdtyUnit	CommodityUnit	30996		MktDef
LctnTyp	LocationType	30283		MktDef
Px	LegPrice	566		SecDef, SecDefUpd
PxDelta	PriceDelta	811		SecDef
PxNotation	PriceNotation	32763		SecDefUpd
PxPrcsn	InstrumentPricePrecision	2576		SecDef,SecDefUpd
PxRngPctg	PriceRangePercentage	2554		MktDef
PxRngProdCmplx	PriceRangeProductComplex	2555		MktDef
PxRngRuleID	PriceRangeRuleID	2556		MktDef, SecDef, SecDefUpd
PxRngRules	PriceRangeRules	Group	Y	MktDef, MktSegGrp
PxRngValu	PriceRangeValue	2553		MktDef
PxTyp	PriceType	423		SecDef, SecDefUpd
QtySclgFctrs	QuantityScalingFactors	Group	Y	MktDef
QtySclgFctr	QuantityScalingFactor	28907		MktDef, SecDef, SecDefUpd
QuotEndTm	QuotingEndTime	30341		SecDef
QuotSideInd	QuoteSideIndicator	2559		MktDef
QuotSideModelTyp	QuoteSideModelType	28898		MktDef
QuotStartTm	QuotingStartTime	30345		SecDef
Qual	InstrumentPartyRoleQualifier	2378		SecDef, SecDefUpd
QuotSzRules	QuoteSizeRules	Group	Y	MktDef, SecDef, SecDefUpd
R	InstrumentPartyRole	1051		SecDef, SecDefUpd
RatioMult	LegRatioMultiplier	28900		SecDef, SecDefUpd
RatioQty	LegRatioQty	623		SecDef, SecDefUpd
RefTickTblID	RefTickTableID	1787		SecDef, SecDefUpd
RelatedMktSeg	RelatedMarketSegments	Group	Y	MktDef
ReltdClsPx	RelatedClosePrice	2589		SecStat
ReltdInstrmt	RelatedInstrumentGrp	Group	Y	SecDef, SecDefUpd
ReltdPx	RelatedPrice	30810		SecDefUpd
ReltdPxTyp	RelatedPriceType	30419		SecDefUpd
RemClndrDays	RemainingCalendarDays	28880		SecStat
RelNdxVal	RelatedIndexValue	28882		SecStat
MinTrdVol	MinTradeVol	562		SecDef
MaxTrdVol	MaxTradeVol	1140		SecDef, SecDefUpd, MktDef
MaxTrdVal	MaxTradeVal	28804		SecDef, SecDefUpd, MktDef
MaxClndrSprdVol	MaxCalendarSpreadVol	28805		MktDef
MaxTESVol	MaxTESVol	28806		MktDef
Rltnshp	MarketSegmentsRelationship	2547		MktDef
RlzdVarnc	RealizedVariance	2587		SecStat
RoundLot	RoundLot	561		SecDef, SecDefUpd

FIXML Attribute	FIX Field Name	FIX Tag / Group	Repeating	FIX Messages
Rsn	SecurityClassificationReason	1583		SecDef
RiskSnsFctr	RiskSensitivityFactor	28904		SecDef
SecClsfnGrp	SecurityClassificationGrp	Group		SecDef
SecRefDataSupplmnt	SecurityReferenceDataSupplement	2962		SecDef, SecDefUpd
SecTrdgRules	SecurityTradingRules	Group	N	SecDef, SecDefUpd
SecTyp	InstrumentScopeSecurityType	1547		MktDef
SecTyp	LegSecurityType	609		SecDef, SecDefUpd
SecTyp	SecurityType	167		SecDef, SecDefUpd, SecStat
SesID	TradingSessionID	336		SecDef
SesSub	TradingSessionSubID	625		SecDef
SetPx	ClearingSettlPrice	2528		SecStat
SetPx	SettlPrice	730		SecStat
SettlBizDays	SettlBusinessDays	28878		SecDef
SettlCcy	SettlCurrency	120		SecDef
SettlMeth	SettlMethod	1193		SecDef,SecDefUpd
SettlSubMeth	SettlSubMethod	2579		SecDef,SecDefUpd
Side	LegSide	624		SecDef, SecDefUpd
Src	InstrumentPartyIDSource	1050		SecDef, SecDefUpd
Src	LegSecurityIDSource	603		SecDef, SecDefUpd
Src	RelatedSecurityIDSource	1651		SecDef, SecDefUpd
Src	SecurityIDSource	22		SecDef, SecDefUpd, SecStat
Src	UnderlyingSecurityIDSource	30305		MktDef
Src	UnderlyingSecurityIDSource	305		SecDef
StartPxRng	StartPriceRange	2551		MktDef
StartTickPxRng	StartTickPriceRange	1206		MktDef
Status	SecurityStatus	965		SecDef, SecDefUpd
StdVarnc	StandardVariance	2588		SecStat
StrkPx	StrikePrice	202		SecDef, SecDefUpd
StrkPxPrcsn	StrikePricePrecision	2577		SecDef,SecDefUpd
SubTyp	SecuritySubType	762		SecDef, SecDefUpd
SvcLctnID1	PrimaryServiceLocationID	2567		MktDef
SvcLctnID2	SecondaryServiceLocationID	2568		MktDef
SvcLctnSubID1	PrimaryServiceLocationSubID	28591		MktDef
SvcLctnSubID2	SecondaryServiceLocationSubID	28593		MktDef
Sym	LegSymbol	600		MktDef
Sym	MarketSegmentSymbol	7177		MktDef
Sym	Symbol	55		MktDef, SecDef, SecDefUpd
Sym	UnderlyingSymbol	311		SecDef
Sym	UnderlyingSymbol	30311		MktDef

FIXML Attribute	FIX Field Name	FIX Tag / Group	Repeating	FIX Messages
TickIncr	TickIncrement	1208		MktDef
TickRuleID	TickRuleID	28887		MktDef
TickRuleProdCmplx	TickRuleProductComplex	2571		MktDef
TickRules	TickRules	Group	Y	MktDef
TotTrdgBizDays	TotalTradingBusinessDays	2585		SecStat
TrdgBizDays	TradingBusinessDays	2586		SecStat
TrdgSesRules	TradingSessionRules	Group	Y	MktSegGrp
TrdgStyle	TradingStyle	31194		SecDefUpd
TslMktGrpID	TslMarketGroupID	28798		MktDef
TslMktGrp	TslMarketGroup	28799		MktDef
TxnTm	TransactTime	60		SecDef, SecDefUpd
Typ	InstrAttribType	871		SecDef, SecDefUpd
Undly		Group	N	MktDef, SecDef
UOM	UnitOfMeasure	996		SecDef, SecDefUpd
UpdActn	SecurityUpdateAction	980		SecDefUpd
USApprvl	USApproval	39543		MktDef
Val	InstrAttribValue	872		SecDef, SecDefUpd
Val	SecurityClassificationValue	1584		SecDef
ValMeth	ValuationMethod	1197		SecDef,SecDefUpd
VegaMult	VegaMultiplier	2583		SecStat
Vol	Volatility	1188		SecStat
VolCorOpnAuct	VolatilityCorridorOpeningAuction	25221		SecDef, SecDefUpd
VolCorIntrAuct	VolatilityCorridorIntradayAuction	25222		SecDef, SecDefUpd
VolCorClsAuct	VolatilityCorridorClosingAuction	25223		SecDef, SecDefUpd
VolCorCont	VolatilityCorridorContinuous	25224		SecDef, SecDefUpd

15 Change log

This document has been through the following iterations:

No	Chapter, page	Date	Change
12.1.1	15, pg. 168	06-Sep-2023	Removed change log for Release 11.0.
12.1.1	Table 9, pg. 27	06-Sep-2023	Updated the Packet header TID for T7 RDI, EMDI, MDI.
12.1.1	11.2.2, pg. 96, 11.2.3, pg. 115	06-Sep-2023	Added new valid value 13 = Non-Standard Volatility Strategy to *ProductComplex (1227)*. Added description for Non-Standard volatility strategies to *SecurityDesc (107)*.
12.1.1	11.2.1, pg. 86	06-Sep-2023	Added new valid value 13 = Non-Standard Volatility Strategy to *TickRuleProductComplex (2571)*, *PriceRangeProductComplex (2555)*, *FlexProductEligibilityComplex (2561)* and *MatchRuleProductComplex (2569)*.
12.1.1	11.3.9, pg. 146	06-Sep-2023	Added new valid value 13 = Non-Standard Volatility Strategy to *ProductComplex (1227)*.
12.1.1	11.3.1, pg. 121	06-Sep-2023	Added new valid value 13 = Non-Standard Volatility Strategy to *ProductComplex (1227)*.
12.1.1	11.3.5, pg. 136,	06-Sep-2023	Added new valid value 13 = Non-Standard Volatility Strategy to *InstrumentScopeProductComplex (1544)*.
12.1.1	11.2.2, pg. 96,	13-Oct-2023	Added new field *SecurityReferenceDataSupplement (2962)* to replace the user defined field *ContractDate (30866)* in future release.
12.1.1	9.4, pg. 61, 11.2.1, pg. 86, 11.3.1, pg. 121, 11.3.2, pg. 129	23-Oct-2023	Remove valid value 50 = Portfolio Compression Trade from *TrdType (828)*.
12.1.1	11.3.1, pg. 121, 11.3.5, pg. 136	06-Nov-2023	Added new valid value 220 = Circuit Breaker Auction Triggered By Static Limit Breach and 221 = Circuit Breaker Auction Triggered By Static Limit Breach Freeze to *SecurityTradingStatus (326)*.
12.1.1	11.2.1, pg. 86, 14.3, pg. 161	06-Nov-2023	Added fields *LocationType (30283)* to mapping table.
12.1.1	11.2.1, pg. 86	11-Nov-2023	Rename field *PowerProductClass (28902)* to *CommodityProductClass (28902)* and *PowerUnitOfMeasure (30996)* to *CommodityUnit (30996)*
12.1.1	11.2.2, pg. 96	21-Nov-2023	Changed FIXML name from *MinTradeVol* to *MinTrdVol*.
12.1.1	11.2.2, pg. 96	21-Nov-2023	Added new fields *MaxTradeVol (1140)* and *MaxTradeVal (28804)* to CashDescriptor.
12.1.1	11.2.1, pg. 86	30-Nov-2023	Added new fields *MaxTradeVol (1140)*, *MaxTradeVal (28804))*, *MaxCalendarSpreadVol (28805)* and *MaxTESVol (28806)*.

No	Chapter, page	Date	Change
12.1.1	11.2.2, pg. 96	31-Jan-2024	Added new field *MaturityFrequencyUnit (2982)* in parallel to user defined field *ContractFrequency (30867)*.
12.1.1	9.10, pg. 70	12-Feb-2024	Removed former section 9.10.2 Market data feed fail-over (T7 EMDI). The section moved to 9.10.2 has been extended to include the information that was referred to before from the removed section. Section 9.10.5 now refers to 9.10.3.
12.1.2	2.5, pg. 11	29-Feb-2024	Changed contact information for Market Supervision (Xetra and Frankfurt).
12.1.2	11.2.2, pg. 96	29-Feb-2024	Modified the descriptions of fields *MaxTradeVol (1140)* and *MaxTradeVal (28804)*.
12.1.2	11.2.5, pg. 118	11-Mar-2024	Added new fields *ConversionMode (28909)* and *ConversionModeQualifier (28910)*. Changed Descriptions for field *RelatedClosePrice (2589)*.
12.1.2	9.8, pg. 66	11-Mar-2024	Removed the sentence "The trade volume record for the future leg of a volatility option strategy is reported without *MDEntryID (278)*".
12.1.3	11.2.2, pg. 96	17-Apr-2024	Changed description of user defined fields *ContractDate (30866)* and *ContractFrequency (30867)* as they continue to be provided alongside to the new fields *SecurityReferenceDataSupplement (2962)* and *MaturityFrequencyUnit (2982)* resp..
12.1.4	11.1.5, pg. 85	06-Jun-2024	Modified paragraph to reflect new technical heartbeat service provided during weekends.
12.1.4	4.3.1, pg. 22, 4.3.2, pg. 22	06-Jun-2024	Paragraph rephrased.
12.1.4	11.1.4, pg. 84, 11.2.1, pg. 86	06-Jun-2024	Added comment about message availability with respect to *MarketSegmentStatus (2542)*.
13.0.1	Table 9, pg. 27	02-Apr-2024	Updated the Packet header TID for T7 EMDI, T7 MDI and T7 RDI.
13.0.1	7.1, pg. 36, 11.1.2, pg. 82	02-Apr-2024	Changed the T7 EMDI Packet Header layout. Increased size of *SenderCompID (49)* to 2 bytes.
13.0.1	11.2.2, pg. 96	02-Apr-2024	Added new optional group *SecurityClassificationGrp*.
13.0.1	11.3.2, pg. 129	02-Apr-2024	Added new optional field *TransBkdTime (483)*.
13.0.1	11.3.2, pg. 129	02-Jun-2024	Added new optional fields *NumberOfBuySides (28911), NumberOfSellSides (28912)*.
13.0.1	11.2.2, pg. 96	14-Jun-2024	Changed *TransactTime (60)* in the context of complex instrument modifications.
13.0.1	11.3.1, pg. 121, 11.3.2, pg. 129	23-Jun-2024	Added new valid value BB = Midpoint Price to *TradeCondition (277)*.
13.0.1	11.2.2, pg. 96	02-Jul-2024	Added new optional fields *MidpointTrading (28913)* and *MidpointExecVenueID (28914)*.
13.0.1	11.3.10, pg. 149	02-Jul-2024	Added mandatory field *ContractDate (30866)*.

No	Chapter, page	Date	Change
13.0.2	11.3.10, pg. 149	19-Aug-2024	Added mandatory field *SecurityReferenceDataSupplement (2962)*.
13.1.1	15, pg. 168	19-Oct-2024	Removed change log for Release 11.1 and 12.0.
13.1.1	2.3.1, pg. 10, 3.4, pg. 17, 4.3.1, pg. 22, 6.2, pg. 31, 6.3, pg. 32, 9.2, pg. 46, 11.2.2, pg. 96, 11.2.3, pg. 115, etc. 11.3.9, pg. 146	19-Oct-2024	Complex and flexible instruments are sent as *Instrument incremental* messages only, even if they existed already on the day before.
13.1.1	14.3, pg. 161	25-Oct-2024	Added missing mapping for *RoundLot (561)*.
13.1.1	11.2.2, pg. 96	04-Nov-2024	Added value HEXP for *ContractDateType (32865)*.
13.1.1	Table 9, pg. 27	06-Sep-2024	Updated the Packet header TID for T7 RDI, EMDI, MDI.
13.1.1	11.2.2, pg. 96, 11.2.3, pg. 115	06-Nov-2024	Added new valid value 14 = Total Return Future Strategy to *ProductComplex (1227)*. Added description for Total Return Future Strategy to *SecurityDesc (107)*.
13.1.1	11.2.1, pg. 86, 11.3.9, pg. 146, 11.3.1, pg. 121, 11.3.5, pg. 136,	06-Nov-2024	Added new valid value 14 = Total Return Future Strategy to *TickRuleProductComplex (2571)*, *PriceRangeProductComplex (2555)*, *FlexProductEligibilityComplex (2561)*, *MatchRuleProductComplex (2569)* and *InstrumentScopeProductComplex (1544)*.
13.1.1	11.2.2, pg. 96	05-Dez-2024	Added new field *PriceNotation (32763)* to *ComplexInstrumentDescriptorGroup*.
13.1.1	11.2.2, pg. 96 11.3.9, pg. 146	12-Dez-2024	Added new fields *RelatedPrice (30810)*, *RelatedPriceType (30419)* and *TradingStyle (31194)* to *InstrmtLegGrp*.
13.1.1	11.2.2, pg. 96	16-Dez-2024	Remove valid value DA = Digital Asset from *SecurityType (167)*.