



FIX-API PLATFORM

FIX 4.4 Taker API Specification

Trade Processor version: 4.5

Document version: 4.5-10

Last revision date: 2025-12-04

CAUTION

We strongly recommend that you try all technical solutions in a demo environment which is as similar as possible to your live server conditions.

We hope you enjoy our product and achieve your exact tasks using it. Our first goal is to make you pleased with the result it brings. Thus, we recommend that you contact our support team to help you install the product and demonstrate its practical use on your demo servers.

Warning

Do not operate the product until you have read and thoroughly understood this document.

CONTENTS

Contents.....	3
Changelog.....	5
Introduction.....	8
FIX sessions.....	8
Viewing FIX session messages.....	12
SSL Encryption.....	13
List of supported FIX messages.....	14
Administrative messages.....	14
Trading session messages.....	15
Market session messages.....	17
Standard message components.....	18
StandardHeader.....	18
StandardTrailer.....	21
Instrument.....	21
OrderQtyData.....	22
Administrative messages.....	23
Logon.....	23
Heartbeat.....	25
TestRequest.....	26
ResendRequest.....	27
Reject.....	28

SequenceReset.....	30
Logout.....	32
Application messages.....	33
Trading session messages.....	33
NewOrderSingle.....	33
ExecutionReport.....	37
OrderCancelRequest.....	43
OrderCancelReject.....	45
OrderStatusRequest.....	47
TradeCaptureReportRequest.....	49
TradeCaptureReportRequestAck.....	51
TradeCaptureReport.....	53
RequestForPositions.....	56
PositionReport.....	57
Market session messages.....	60
MarketDataRequest.....	60
MarketDataSnapshotFullRefresh.....	64
MarketDataIncrementalRefresh.....	66
MarketDataRequestReject.....	70
Other messages.....	72
BusinessMessageReject.....	72
Contact us.....	74

CHANGELOG

Revision date	Doc version	TP version	Changes
2025-12-04	4.5-10	4.5	<ul style="list-style-type: none">• No changes to the FIX protocol.• Updated TP version.
2025-10-10	4.4-10	4.4	<ul style="list-style-type: none">• No changes to the FIX protocol.• Updated TP version.
2025-07-30	4.3-10	4.3	<ul style="list-style-type: none">• No changes to the FIX protocol.• Updated TP version.
2025-06-05	4.2-20	4.2	<ul style="list-style-type: none">• No changes to the FIX protocol.• Updated document design.• Minor wording and document structure fixes.
2025-05-01	4.2-10	4.2	<ul style="list-style-type: none">• No changes to the FIX protocol.• Updated TP version.
2025-02-26	4.1-20	4.1	<ul style="list-style-type: none">• No changes to the FIX protocol.• Added additional description to MarketDataRequest message with tag SubscriptionRequestType(263) = 2. This description is relevant for all TP versions.
2025-01-31	4.1-10	4.1	<ul style="list-style-type: none">• No changes to the FIX protocol, however now, TP also supports string account logins

			<p>in Account(1).</p> <ul style="list-style-type: none"> Updated TP version.
2024-10-11	4.0-10	4.0	<ul style="list-style-type: none"> No changes to the FIX protocol. Updated TP version.
2024-06-28	3.5.7-10	3.5.7	<ul style="list-style-type: none"> No changes to the FIX protocol. Updated TP version. Updated the FIX session fields name/description for Password and Platform symbols and tags allowed for feeding fields.
2024-06-18	3.5.6-11	3.5.6	<ul style="list-style-type: none"> No changes to the FIX protocol. Fixed the incorrect description for the TimeInForce(59) required parameter from the NewOrderSingle message. This field is and always was considered as required.
2024-04-26	3.5.6-10	3.5.6	<ul style="list-style-type: none"> No changes to the FIX protocol. Updated TP version.
2024-03-29	3.5.5-10	3.5.5	<ul style="list-style-type: none"> No changes to the FIX protocol. Updated TP version.
2024-02-09	3.5.4-10	3.5.4	<ul style="list-style-type: none"> No changes to the FIX protocol. Updated TP version.
2023-12-20	3.5.3-10	3.5.3	<ul style="list-style-type: none"> No changes to the FIX protocol. Updated TP version.

2023-10-13	3.5.1-20	3.5.1	<ul style="list-style-type: none"> • No changes to the FIX protocol. • Fixed document layout.
2023-10-04	3.5.1-10	3.5.1	<ul style="list-style-type: none"> • No changes to the FIX protocol. • Added descriptions of Save quotes depth and Feeding only session session parameters.
2023-08-23	3.5.0-10	3.5.0	<ul style="list-style-type: none"> • No changes to the FIX protocol. • Updated TP version.
2023-03-04	3.4.1-10	3.4.1	<ul style="list-style-type: none"> • No changes to the FIX protocol. • Updated TP version.
2022-12-09	3.4.0-10	3.4.0	<ul style="list-style-type: none"> • No changes to the FIX protocol. • Updated TP version. • Added a note that the password passed in Password(554) from the Logon message is replaced with ***** value in logs.
2022-11-11	3.3.3-15	3.3.3	<ul style="list-style-type: none"> • No changes to the FIX protocol. • Added FIX sessions section. • Updated links to platform and session creation steps. • Updated SSL connection settings description.
2022-11-09	3.3.3-14	3.3.3	<ul style="list-style-type: none"> • No changes to the FIX protocol. • Minor fixes.
2022-11-01	3.3.3-13	3.3.3	<ul style="list-style-type: none"> • No changes to the FIX protocol.

			<ul style="list-style-type: none"> Added an example of how market data is updated based on a MarketDataIncrementalRefresh message.
2022-10-30	3.3.3-12	3.3.3	<ul style="list-style-type: none"> No changes to the FIX protocol. Added the Changelog section to the document.
2022-10-28	3.3.3-11	3.3.3	<ul style="list-style-type: none"> No changes to the FIX protocol. Updated TP version.
2022-10-27	3.3.1-10	3.3.3	<ul style="list-style-type: none"> Initial version.

INTRODUCTION

This document outlines the FIX 4.4 Taker API engagement rules for FIX clients connecting to the **FIX-API platform** of Trade Processor (TP).

FIX clients can connect to both Market and Trading sessions via this API.

FIX SESSIONS

Trade Processor's **FIX-API platform** allows the connection of multiple FIX sessions simultaneously.

Each of the created FIX sessions is able to support both Market and Trading FIX messages, however, it is strongly advised to create separate FIX sessions for Market data and Trading data to avoid any potential delays in message exchange.

A description of the **FIX-API platform**, as well as the ways to create FIX sessions can be found in the corresponding Trade Processor User Guide section.

How to create a FIX session:

- Create a FIX-API platform (User guide section: [FIX-API platform > FIX-API platform setup](#)).
- Configure the necessary settings of your FIX-API platform:
 - Create FIX-API accounts (User guide section: [FIX-API platform > FIX-API accounts](#)).
 - Create FIX-API platform symbols (User guide section: [FIX-API platform > FIX-API symbols](#)).
 - Set up swaps (User guide section: [FIX-API platform > Swaps](#)).
- Create the FIX sessions (User guide section: [FIX-API platform > Manage FIX sessions](#)).

Each FIX session contains the following configurable parameters:

- **FIX server name** – name of the receiver of FIX messages. This value must be specified in the **TargetCompID** field of a FIX message to connect to this session.

Note: This field is uneditable after saving.

- **Client session ID** – name of the sender of FIX messages. This value must be specified in the **SenderCompID** field of a FIX message to connect to this session.

Note: This field is uneditable after saving.

- **Begin string** – type of FIX protocol. Set to **FIX.4.4** by default.

- **Username** – username for connection via this FIX session..

- **Password** – password for connection via this FIX session.

- **Sessions default account** – default trading account. If no trading account is provided in a FIX message, the trading operation will be associated with this default account.

Note: this account must be included in the [Accounts allowed for session](#) and [Groups allowed for session](#) fields' values.

- **T4B Fix-protocol version** – order volume conversion parameter that enables compatibility with Trade Processor versions before 3.2 if Trade Processor is used as a FIX-client.

- **FIXAPI 3.2** – volumes are transmitted via this FIX session in base units.

- **FIXAPI 3.1** – FIX-API platform expects that volumes are transmitted in lots (where 1 lot = 100 000 000) and converts transmitted volumes as described below:

- Volumes received via **Trading session** are converted to base units using the following formula:

Volume in TP base units = Volume in FIX adj. units * CS / 100 000 000

- Volumes sent out via **Market session** are converted from base units using the following formula:

Volume in FIX adj. units = Volume in TP base units * 100 000 000 / CS

Where **CS** is the **Contract size** setting of a FIX-API platform symbol.

Tip: Choose this option when the version of Trade Processor that connects to the FIX-API platform is 3.2 or older.

- **Save quotes depth** – setting that defines whether to save quotes fed by the FIX-API platform to a separate database and the depth of market to be saved.
 - **Nothing** – no market data is saved to the database.
 - **From 1 to 10** – the number of levels to be saved to the database.
- **Accounts allowed for session** – trading accounts that are allowed to trade via this FIX session. To fill in this field, masks with ! and * can be used:
 - * matches any symbol or sequence of symbols. For example, if **100*** is specified, then any account starting with 100 matches.
 - ! excludes the value it precedes. For example, if **!1000** is specified, then account 1000 is excluded.
- **Groups allowed for session** – groups that are allowed to trade via this FIX session. To fill in this field, masks with ! and * can be used:
 - * matches any symbol or sequence of symbols. For example, if **T4B*** is specified, then any group starting with T4B matches.
 - ! excludes the value it precedes. For example, if **!T4B*** is specified, then all groups starting with T4B are excluded.

Note: **Accounts allowed for session** and **Groups allowed for session** follow the **AND** logic. This means that a trading account of an order must be included in the **Accounts allowed for session** and **Groups allowed for session** fields' values.

- **Symbols and tags allowed for feeding** – specifies FIX-API platform symbols and/or their tags for which the quotes are sent for the chosen feeding session. Supports the use of masks (e.g. * or !). By default is set to * (all FIX-API platform symbols) and can only be modified to a custom value if the [Feeding only session](#) checkbox is enabled.
- **Ignore log options** – defines FIX message types that are not logged. The following options are available:
 - **Nothing** – all received messages are logged.
 - **Quote** – quote messages are not logged.
 - **Other** – all messages except for quotes are not logged.
- **Feeding only session** – if enabled, FIX-API platform will only be able to send Market data through this FIX session.

VIEWING FIX SESSION MESSAGES

FIX session messages are stored in the Trade Processor Gateway directory at the following path:

TradeProcessor\Gateway\Logs\Connectors\{FIX-API platform name}\

There are two types of **.log** files stored in this directory:

- **Message logs** – contain the FIX messages themselves:

FIX.4.4-{FIX server}-{Client session ID} {YYYY-MM-DD}.messages.current.log

- **Event logs** – contain the FIX events that correspond to the FIX messages:

FIX.4.4-{FIX server}-{Client session ID} {YYYY-MM-DD}.events.current.log

SSL ENCRYPTION

You can enable the SSL encryption of the data transmitted through the FIX sessions of a FIX-API platform.

SSL encryption is set up in the Trade Processor Web interface at [Platforms > FIX-API platform > Settings > SSL settings](#).

The following settings are available:

- **SSL enable** – defines whether to disable or enable the SSL encryption.
- **SSL certificate** – path to the FIX server SSL certificate.
- **SSL certificate password** – password to the FIX server SSL certificate.
- **SSL CA certificate** – path to the FIX client root SSL certificate.
- **SSL server name** – certificate name of the server.
- **SSL protocols** – version of the used SSL protocol.
- **SSL check certificate revocation** – defines whether to check the client SSL certificate against revocation lists.
- **SSL require client certificate** – defines whether to require a FIX client SSL certificate.
- **SSL validate certificate** – defines whether to validate a FIX client SSL certificate. Validation is performed using the root certificate specified in the **SSL CA certificate** field.

Note: The SSL parameters are optional and can be omitted if the SSL encryption is not required.

LIST OF SUPPORTED FIX MESSAGES

This section lists all the FIX message types supported by the **FIX-API platform** of Trade Processor (TP).

ADMINISTRATIVE MESSAGES

Table 1. List of supported administrative FIX message types

Message type	Message name	Description
A	Logon	Message used to establish a connection between a FIX client and TP.
0	Heartbeat	Message sent during quiet intervals to ensure that connection is still alive.
1	TestRequest	Message used to request a Heartbeat message from the counterparty to allow testing connection.
2	ResendRequest	Message used to request from the counterparty to resend a range of messages.
3	Reject	Message sent in response to a message which is structurally invalid (i.e. invalid or missing tag or ill-formed tag value).
4	SequenceReset	Either replaces administrative messages or resets message sequence.

5	Logout	Message used to terminate a FIX connection. Can be sent by a FIX client or TP.
---	------------------------	--

TRADING SESSION MESSAGES

Table 2. List of supported Trading session FIX message types

Message type	Message name	Description
D	NewOrderSingle	Message sent by a FIX client to place a new order.
8	ExecutionReport	Message sent by TP to report the change in status of a FIX client order.
F	OrderCancelRequest	Message sent by a FIX client to cancel an existing order.
9	OrderCancelReject	Message sent by TP when an OrderCancelRequest message cannot be successfully processed.
H	OrderStatusRequest	Message sent by a FIX client to request the state of an order.
AD	TradeCaptureReportRequest	Message sent by a FIX client in order to request historic trades.

Message type	Message name	Description
AQ	TradeCaptureReportRequestAck	Message sent by TP in response to a TradeCaptureReportRequest message to inform whether the request was successful.
AE	TradeCaptureReport	Message sent by TP with trade information.
AN	RequestForPositions	Message sent by a FIX client to request positions for a specified account.
AP	PositionReport	Message sent by TP with information on found positions.

MARKET SESSION MESSAGES

Table 3. List of supported Market session FIX message types

Message type	Message name	Description
V	MarketDataRequest	Message sent by a FIX client to request market data.
W	MarketDataSnapshotFullRefresh	Message sent by TP with initial market data snapshot. This message is also sent to update market data.
X	MarketDataIncrementalRefresh	Message sent by TP for incremental updates of market data. Contains a combination of new, changed, or deleted market data entries.
Y	MarketDataRequestReject	Message sent by TP to inform that the market data request was rejected.

STANDARD MESSAGE COMPONENTS

This section describes the standard components of all FIX messages supported by the **FIX-API platform** of Trade Processor (TP).

Further in the document, descriptions of the messages that contain any of these components refer to this section.

STANDARDHEADER

Table 1. Standard Header components of the FIX messages

Tag	Tag name	Description	Required	Data type
8	BeginString	The FIX version number identifier. Always set as FIX.4.4	Y	String
9	BodyLength	Length of the FIX message.	Y	Length
35	MsgType	The type of the message.	Y	String
34	MsgSeqNum	The unique sequence number for the message within the current FIX session. Note: Messages from a FIX client to TP and messages from TP to a FIX client belong to different sequences.	Y	SeqNum

Tag	Tag name	Description	Required	Data type
49	SenderCompID	<p>Identifier for the sender of the message.</p> <p>For messages sent from a FIX client to TP, it is the value of the Client session ID setting of the FIX session.</p> <p>For messages sent from TP to a FIX client, it is the value of the FIX server name setting of the FIX session.</p>	Y	String
52	SendingTime	<p>Time at which the message was sent.</p> <p>Time and date represented in UTC in YYYYMMDD-HH:MM:SS.sss format.</p> <p>Example: 20221004-04:36:53.903</p>	Y	UTCTimestamp
56	TargetCompID	<p>Identifier for the receiver of the message.</p> <p>For messages sent from a FIX client to TP, it is the value of the FIX server name setting of the FIX session.</p> <p>For messages sent from TP to a FIX client, it is the value of the Client session ID setting of the FIX session.</p>	Y	String
43	PossDupFlag	Indicates that the message with this sequence number is possibly retransmitted.	N	Boolean

Tag	Tag name	Description	Required	Data type
		<p>Required for re-sent messages, whether resending is prompted by the sending system or as the result of a resend request (ResendRequest message):</p> <ul style="list-style-type: none"> • Y – possible duplicate. • N – original transmission. <p>Set to Y if a message is re-sent.</p>		
97	PossResend	Never used.	N	Boolean
122	OrigSendingTime	<p>Date and time when the message was originally sent. Required for a message resulting from a resend request.</p> <p>Time and date represented in UTC in YYYYMMDD-HH:MM:SSsss format.</p>	N	UTCTimestamp

STANDARDTRAILER

Table 2. Standard Trailer components of the FIX messages

Tag	Tag name	Description	Required	Data type
10	CheckSum	Checksum for message contents.	Y	String

INSTRUMENT

Table 3. Standard Instrument components of the FIX messages

Tag	Tag name	Description	Required	Data type
55	Symbol	FIX-API platform symbol name.	C *	String

* **Note:** Instrument component may not be required for specific FIX message types. The specific requirements for this component are listed in each corresponding FIX message description further in the document.

ORDERQTYDATA

Table 4. Standard Order Qty Data components of the FIX messages

Tag	Tag name	Description	Required	Data type
38	OrderQty	Order quantity in base units, unless T4B Fix-protocol version session setting in TP is set to FIXAPI 3.1 .	Y	Qty

ADMINISTRATIVE MESSAGES

This section describes the Administrative FIX messages supported by the **FIX-API platform** of Trade Processor (TP).

LOGON

Logon is the first message a FIX client should send to TP after establishing a TCP connection with it. The FIX client should then wait for a **Logon** message from TP before sending any other messages to TP.

Table 1. Logon FIX message components

Tag	Tag name	Description	Required	Data type
	StandardHeader		Y	
	MsgType(35) = A			
98	EncryptMethod	Method of encryption. Supported value – 0 (None).	Y	Int
108	HeartBtInt	Maximum time (seconds) to wait before a Heartbeat message is sent in the absence of other activity on the connection. The same value is used by both parties.	Y	Int

Tag	Tag name	Description	Required	Data type
141	ResetSeqNumFlag	<p>Indicates whether both sides of the FIX session should reset sequence numbers on logon:</p> <ul style="list-style-type: none"> ● Y – both sides reset sequence numbers. ● N – both sides continue the previous numeration before logon. 	N	Boolean
383	MaxMessageSize	Maximum number of bytes supported for a single message.	N	Length
553	Username	<p>FIX session username, specified in the Username field of the FIX session settings of the FIX-API platform.</p> <p>Required, when a Logon message is sent by a FIX client.</p>	N	String
554	Password	<p>Fix session password or passphrase, specified in the Password field of the FIX session settings of the FIX-API platform.</p> <p>Required, when a Logon message is sent by a FIX client.</p> <p>Note: In FIX session logs, this password is replaced with ***** value for safety reasons.</p>	N	String
<u>StandardTrailer</u>			Y	

HEARTBEAT

Heartbeat is a message sent during quiet intervals to ensure that the connection between a FIX client and TP is still alive.

The frequency of **Heartbeat** messages is set in the **Logon** message with the **HeartBtInt (108)** parameter.

This type of message is also sent in response to a **TestRequest** message.

Table 2. Heartbeat FIX message components

Tag	Tag name	Description	Required	Data type
		StandardHeader		
		MsgType(35) = 0	Y	
112	TestReqID	Identifier from a TestRequest message. Required when a Heartbeat message is sent in response to a TestRequest message.	N	String
		StandardTrailer	Y	

TESTREQUEST

TestRequest is a message used to request a **Heartbeat** message from the counterparty to test the connection.

Table 3. TestRequest FIX message components

Tag	Tag name	Description	Required	Data type
	StandardHeader		Y	
	MsgType(35) = 1			
112	TestReqID	Identifier to be returned in a Heartbeat message.	Y	String
	StandardTrailer		Y	

RESENDREQUEST

ResendRequest is a message used to request re-sending the FIX messages from a counterparty. In response, the counterparty re-sends the FIX messages with **MsgSeqNum(34)** within the specified range.

If the specified range includes Administrative messages, then they are replaced with a **SequenceReset** message with **GapFillFlag(123) = Y**.

Table 4. ResendRequest FIX message components

Tag	Tag name	Description	Required	Data type
	StandardHeader		Y	
	MsgType(35) = 2			
7	BeginSeqNo	MsgSeqNum(34) of the first message in the range to be resent.	Y	SeqNum
16	EndSeqNo	MsgSeqNum(34) of the last message in the range to be resent. To request a single message, EndSeqNo should be equal to BeginSeqNo . To request all messages subsequent to a particular message, EndSeqNo should be set to 0 (where 0 represents infinity).	Y	SeqNum
	StandardTrailer		Y	

REJECT

Reject is a message sent in response to a FIX message that is structurally invalid (e.g. where a tag / tag value is invalid or a required tag / tag value is missing) and therefore cannot be properly processed.

Table 5. Reject FIX message components

Tag	Tag name	Description	Required	Data type
	StandardHeader		Y	
	MsgType(35) = 3			
45	RefSeqNum	MsgSeqNum(34) of the rejected message.	Y	SeqNum
371	RefTagID	The tag number of the FIX field being referenced.	N	Int
372	RefMsgType	The MsgType(35) of the FIX message being referenced.	N	String
373	SessionRejectReason	Code to identify the reason for rejection: <ul style="list-style-type: none"> • 0 – invalid tag number. • 1 – required tag missing. • 2 – tag not defined for this message type. • 3 – undefined tag. 	N	Int

Tag	Tag name	Description	Required	Data type
		<ul style="list-style-type: none"> ● 4 – tag specified without a value. ● 5 – value is incorrect (out of range) for this tag. ● 6 – incorrect data format for value. ● 7 – decryption problem. ● 8 – signature problem. ● 9 – ComplID problem (SenderComplID, TargetComplID, or both). ● 10 – sendingTime accuracy problem. ● 11 – invalid MsgType. ● 13 – tag appears more than once (non-repeating group tags only). ● 14 – tag specified out of required order. ● 15 – repeating group fields out of order. ● 99 – other. 		
58	Text	Where possible, a message to explain the reason for rejection.	N	String
StandardTrailer			Y	

SEQUENCERESET

SequenceReset is a message used in two modes:

- **Sequence reset mode** – a message is sent as a request to reset a message sequence. In this case, **GapFillFlag(123)** is set to N.
- **Gap filling mode** – a message is sent in response to a [ResendRequest](#) message to replace administrative messages (as TP does not re-send administrative messages). In this case, **GapFillFlag(123)** is set to Y.

A **SequenceReset** message fills the gap in the re-sent message sequence, i.e. it replaces an administrative message or a group of consequential administrative messages. In this case:

- **MsgSeqNum(34)** specifies the next expected sequence number.
- **NewSeqNo(36)** specifies the highest sequence number in the group of administrative messages plus 1.
- **GapFillFlag(123)** is set to Y.

Example: In the range of messages requested for retransmission there is a group of consequential administrative messages with sequence numbers from 4 to 10. In this case, a single **SequenceReset** message is sent to replace this group with **MsgSeqNum(34) = 4** (because it is the next expected sequence number in the range of messages to resent) and **NewSeqNo(36) = 11** (because it is the next number to be transmitted after the last administrative message in the group).

Table 6. SequenceReset FIX message components

Tag	Tag name	Description	Required	Data type
	StandardHeader	In the StandardHeader component, PossDupFlag(43) : <ul style="list-style-type: none"> Always required when GapFillFlag(123) is set to Y. Required when GapFillFlag(123) is set to N and the new sequence overlaps the previous sequence (i.e. NewSeqNo is less than the last MsgSeqNum in the previous sequence). 	Y	
123	GapFillFlag	Indicates in which mode the SequenceReset message is used: <ul style="list-style-type: none"> Y – Gap Fill mode. MsgSeqNum(34) field is valid. N – Sequence Reset mode. MsgSeqNum(34) is ignored. 	N	Boolean
36	NewSeqNo	New sequence number.	Y	SeqNum
	StandardTrailer		Y	

LOGOUT

Logout is a message used to terminate a FIX connection. Can be sent by a FIX client or TP.

In response, a counterparty also sends a **Logout** message.

Table 7. Logout FIX message components

Tag	Tag name	Description	Required	Data type
		StandardHeader		
		MsgType(35) = 5	Y	
58	Text	If TP logs out a FIX client, this field can contain a text description with the logout reason.	N	String
		StandardTrailer	Y	

APPLICATION MESSAGES

This section describes the Application FIX messages supported by the **FIX-API platform** of Trade Processor (TP).

Some Application messages contain repeating tag groups. A repeating group consists of a set of tags that repeat in a FIX message, and it starts with a tag that specifies the number of the repeated sets.

Example: There is **NoPositions** repeating group, which consists of **NoPositions**, **PosType**, **LongQty**, and **ShortQty** tags. Therefore, if in a FIX message **NoPositions = 2**, then this repeating group in the message will look as follows: **NoPositions = 2 PosType LongQty ShortQty PosType LongQty ShortQty**.

For each of the described Application messages below, the tags that repeat are marked as **Repeating tags**.

TRADING SESSION MESSAGES

NewOrderSingle

NewOrderSingle is a message sent by a FIX client to place a new order.

In response, TP sends an **ExecutionReport** message(s) to confirm that an order was received with **OrdStatus(39) = 0 (New)** followed by **ExecutionReport** messages informing of the order execution result.

If there were multiple fills of an order, then for each fill TP sends an execution report.

Example: an order is sent to a liquidity pool with two liquidities and executed on both liquidities. FIX-client receives:

- Two **ExecutionReport** messages with **OrdStatus(39) = 0** (New) to confirm that each part of the order was accepted.
- One **ExecutionReport** message with **OrdStatus(39) = 1** (Partially filled) to confirm the execution of a part of the order on one liquidity.
- One **ExecutionReport** message with **OrdStatus(39) = 2** (Filled) to confirm the execution of the residual part of the order on the other liquidity.

Table 1. NewOrderSingle FIX message components

Tag	Tag name	Description	Required	Data type
	StandardHeader		Y	
	MsgType(35) = D			
11	ClOrdID	Identifier for the order supplied by the FIX client. ClOrdId is restricted to a maximum of 40 arbitrary ASCII characters.	Y	String
	Instrument		Y	
1	Account	<p>The FIX-API platform account for which the order should be placed. If not specified, then the platform default account is used.</p> <p>Note: This value must be specified in Accounts allowed for session (FIX session setting in TP) AND belong to a group from Groups allowed for session (FIX session setting in TP), otherwise, the order is rejected.</p>	N	String

Tag	Tag name	Description	Required	Data type
		Additionally, this value is used to route the order based on the Logins parameter of the FIX-API platform Routing rules in Execution configuration .		
54	Side	<p>Side of the order:</p> <ul style="list-style-type: none"> ● 1 – BUY. ● 2 – SELL. 	Y	Char
60	TransactTime	<p>Time at which this order request was initiated/released by the trader, trading system, or intermediary.</p> <p>Time and date represented in UTC in YYYYMMDD-HH:MM:SSsss format.</p>	Y	UTCTimestamp
<u>OrderQtyData</u>				Y
40	OrdType	<p>Order type:</p> <ul style="list-style-type: none"> ● 1 – Market. ● 2 – Limit. 	Y	Char
44	Price	Order price. Required only for Limit orders (where OrdType(40) = 2).	N	Price
59	TimeInForce	Specifies how long the order remains in effect:	Y	Char

Tag	Tag name	Description	Required	Data type
		<ul style="list-style-type: none"> ● 1 – Good Till Cancelled (GTC). ● 3 – Immediate or Cancel (IOC). ● 4 – Fill or Kill (FOK). <p>For market orders (where OrdType(40) = 1), only values of 3 and 4 are valid.</p> <p>For limit orders (where OrdType(40) = 2), all values are valid.</p>		
	StandardTrailer		Y	

ExecutionReport

ExecutionReport is a message sent by TP that reports a change in status of the FIX client's order.

This message is sent in order to:

- Confirm that an order was accepted by TP (in response to **NewOrderSingle** message).
- Reject an order or its part (in response to **NewOrderSingle** message).
- Confirm that the order is cancelled (in response to **OrderCancelRequest**).
- Provide execution information on existing orders after confirmation that the order was accepted. If the order was executed in parts, then for each filled part, a separate **ExecutionReport** message is sent.
- Provide order status information (in response to **OrderStatusRequest** message).

Table 2. ExecutionReport FIX message components

Tag	Tag name	Description	Required	Data type
	StandardHeader		Y	
	MsgType(35) = 8			
37	OrderID	Unique identifier allocated by TP for the order whose execution status is being reported.	Y	String
11	ClOrdID	Identifier for the order supplied by the FIX client whose execution status is being reported (from the NewOrderSingle message).	N	String

Tag	Tag name	Description	Required	Data type
		If the ExecutionReport message is sent in response to an OrderCancelRequest message, it is the ClOrdID of the OrderCancelRequest message. TP always fills this field if it was provided in the NewOrderSingle message.		
41	OrigClOrdID	Required if the ExecutionReport message is a response to OrderCancelRequest message to specify ClOrdID(11) of the order that was cancelled, i.e. ClOrdID(11) from the NewOrderSingle message.	N	String
790	OrdStatusReqID	Identifier (OrdStatusReqID) of an OrderStatusRequest message. This field is only returned if OrdStatusReqID was provided in the OrderStatusRequest message.	N	String
17	ExecID	The unique identifier allocated by TP for this ExecutionReport message.	Y	String
150	ExecType	The reason for sending this ExecutionReport message: <ul style="list-style-type: none"> • 0 – New (Order or its part placed). • 4 – Cancelled (Order or its part cancelled). • 8 – Rejected (Order or its part rejected). • F – Trade (Order or its part executed). 	Y	Char

Tag	Tag name	Description	Required	Data type
39	OrdStatus	<p>Current state of the order:</p> <ul style="list-style-type: none"> • 0 – New (Order placed but not executed). • 1 – Partially filled (Order with executions and remaining quantity). • 2 – Filled (Order completely filled with no remaining quantity). • 4 – Cancelled (Order cancelled). • 8 – Rejected (Order rejected). 	Y	Char
103	OrdRejReason	<p>Only for messages with ExecType(150) = 8 (Rejected). Contains the reason for rejection:</p> <ul style="list-style-type: none"> • 0 – Broker / Exchange option. • 1 – unknown symbol. • 2 – Exchange closed. • 3 – order exceeds limit. • 4 – too late to enter. • 5 – unknown order. • 6 – duplicate order (e.g. CIOrdID(11)). • 7 – duplicate of a verbally communicated order. • 8 – stale order. 	N	Int

Tag	Tag name	Description	Required	Data type
		<ul style="list-style-type: none"> ● 9 – trade along required. ● 10 – invalid Investor ID. ● 11 – unsupported order characteristic. ● 12 – surveillance option. ● 13 – incorrect quantity. ● 14 – incorrect allocated quantity. ● 15 – unknown account(s). ● 99 – other. 		
1	Account	A FIX-API platform account on behalf of which the order was placed (Account(1) from the NewOrderSingle message) .	N	String
	<u>Instrument</u>			Y
54	Side	Side of the order: <ul style="list-style-type: none"> ● 1 – BUY. ● 2 – SELL. 	Y	Char
	<u>OrderQtyData</u>			Y

Tag	Tag name	Description	Required	Data type
40	OrdType	Order type: <ul style="list-style-type: none">• 1 – Market.• 2 – Limit.	N	Char
44	Price	Order price. Present, if specified in the order whose status is being reported.	N	Price
59	TimeInForce	Specifies how long the order remains in effect: <ul style="list-style-type: none">• 1 – Good Till Cancelled (GTC).• 3 – Immediate or Cancel (IOC).• 4 – Fill or Kill (FOK).	N	Char
32	LastQty	Order volume executed on this fill: <ul style="list-style-type: none">• In base units if T4B Fix-protocol version session setting in TP is FIXAPI 3.2.• In the same units as received in the NewOrderSingle message if T4B Fix-protocol version session setting in TP is FIXAPI 3.1. <p>Only for messages with ExecType(150) = F (Trade).</p>	N	Qty
31	LastPx	Price of the last (this) fill.	N	Price

Tag	Tag name	Description	Required	Data type
151	LeavesQty	<p>Quantity open for further execution:</p> <ul style="list-style-type: none"> • In base units if T4B Fix-protocol version session setting in TP is FIXAPI 3.2. • In the same units as received in the NewOrderSingle message if T4B Fix-protocol version session setting in TP is FIXAPI 3.1. 	Y	Qty
14	CumQty	<p>Total executed quantity of the reported order:</p> <ul style="list-style-type: none"> • In base units if T4B Fix-protocol version session setting in TP is FIXAPI 3.2. • In the same units as received in the NewOrderSingle message if T4B Fix-protocol version session setting in TP is FIXAPI 3.1. 	Y	Qty
6	AvgPx	Average price of all fills of the order.	Y	Price
60	TransactTime	<p>The time of the transaction represented by this ExecutionReport message.</p> <p>Time and date represented in UTC in YYYYMMDD-HH:MM:SSsss format.</p>	N	UTCTimestamp
58	Text	Free format text string. This field does not have a specified maximum length.	N	String
<u>StandardTrailer</u>			Y	

OrderCancelRequest

OrderCancelRequest is a message sent by a FIX client to cancel an existing order.

Only orders whose current **OrdStatus(39)** is **0** (New) or **1** (Partially filled) can be cancelled.

In response, TP either sends an **ExecutionReport** message with **OrdStatus(39) = 4** to confirm cancellation or an **OrderCancelReject** message to reject the request.

Table 3. OrderCancelRequest FIX message components

Tag	Tag name	Description	Required	Data type
	StandardHeader		Y	
	MsgType(35) = F			
41	OrigClOrdID	ClOrdID(11) of the order to cancel (from the original NewOrderSingle message).	Y	String
1	Account	A FIX-API platform account who attempts to cancel the order.. If not specified, then the platform default account is used.	N	String
11	ClOrdID	The FIX-client-supplied identifier assigned to the OrderCancelRequest message. ClOrdId is restricted to a maximum of 40 arbitrary ASCII characters.	Y	String

Tag	Tag name	Description	Required	Data type
		<u>Instrument</u>	Y	
54	Side	Side of the order: <ul style="list-style-type: none">● 1 – BUY.● 2 – SELL.	Y	Char
60	TransactTime	The time at which this request was initiated by a FIX client.. Time and date represented in UTC in YYYYMMDD-HH:MM:SSsss format.	Y	UTCTimestamp
		<u>OrderQtyData</u>	Y	
		<u>StandardTrailer</u>	Y	

OrderCancelReject

OrderCancelReject is a message sent by TP when an **OrderCancelRequest** message cannot be successfully processed.

Table 4. OrderCancelReject FIX message components

Tag	Tag name	Description	Required	Data type
		<u>StandardHeader</u> MsgType(35) = 9	Y	
37	OrderID	The OrderId(37) of the order that failed to be cancelled (from the ExecutionReport message). If CxlRejReason(102)= 1 (Unknown), then OrderId = NONE .	Y	String
11	ClOrdID	ClOrdID(11) of the OrderCancelRequest message for which rejection is received.	Y	String
41	OrigClOrdID	ClOrdID(11) of the order whose cancellation is rejected (from the NewOrderSingle message with which the order was placed).	Y	String
39	OrdStatus	OrdStatus(39) after this cancel reject is applied. If CxlRejReason(102) = 1 (Unknown), then OrdStatus(39) = 8 (Rejected) .	Y	Char

Tag	Tag name	Description	Required	Data type
1	Account	The FIX-API platform account on behalf of which the OrderCancelRequest message was sent.	N	String
434	CxlRejResponseTo	Identifies the type of request that is rejected: Supported value – 1 (OrderCancelRequest).	Y	Char
102	CxlRejReason	Code to identify the reason for cancel request rejection: <ul style="list-style-type: none"> ● 1 – unknown order. ● 2 – Broker / Exchange Option. ● 99 – other. 	N	Int
58	Text	Free format text string. Can be used to specify details of rejection.	N	String
<u>StandardTrailer</u>				Y

OrderStatusRequest

OrderStatusRequest is a message sent by a FIX client to request the current state of an order.

In response, TP sends an **ExecutionReport** message with the most recent order status.

Example: An order was partially filled (there were two execution reports: the first with **OrdStatus(39) = 1** (Partially filled) and the second with **OrdStatus(39) = 4** (Cancelled)). The **OrdStatus(39)** from the **ExecutionReport** response message is **4** (Cancelled), as it is the most recent status (i.e. execution report with Cancelled status was sent after the execution report with Partial Filling status).

In an **ExecutionReport** message sent in response to an **OrderStatusRequest** message:

- **ExecID(17) = 0**.
- **TransactTime(60)** is the date and time of the **OrderStatusRequest** message.

Note: Orders that were rejected for whatever reason will be reported as Unknown.

Table 5. OrderStatusRequest FIX message components

Tag	Tag name	Description	Required	Data type
	StandardHeader		Y	
	MsgType(35) = H			
11	CIOrdID	CIOrdID(11) of the order whose status is being requested (from the NewOrderSingle message with which the order was placed).	Y	String

Tag	Tag name	Description	Required	Data type
1	Account	The FIX-API platform account with which the order whose state is being requested was placed.	N	String
790	OrdStatusReqID	<p>Optional, can be used to uniquely identify a specific OrderStatusRequest message.</p> <p>If provided, returned in the response ExecutionReport message.</p> <p>Restricted to a maximum of 16 hexadecimal characters with no leading 0 characters (essentially this is a 64-bit number).</p>	N	String
<u>Instrument</u>				Y
54	Side	<p>Side of the order:</p> <ul style="list-style-type: none"> ● 1 – BUY. ● 2 – SELL. 	Y	Char
<u>StandardTrailer</u>				Y

TradeCaptureReportRequest

TradeCaptureReportRequest is a message sent by a FIX client to request historic trades within the specified time period.

In response, TP sends a **TradeCaptureReportRequestAck** message to provide the result of request.

Further, if any trades are found within the specified period, **TradeCaptureReport** messages are sent with trade data. One **TradeCaptureReport** message represents one trade (i.e. the corresponding **ExecutionReport** message has **ExecType(150) = F (Trade)**).

Table 6. TradeCaptureReportRequest FIX message components

Tag	Tag name	Description	Required	Data type
	StandardHeader		Y	
	MsgType(35) = AD			
568	TradeRequestID	Identifier of TradeCaptureReportRequest message.	Y	String
569	TradeRequestType	<p>Type of request.</p> <p>Supported value – 1 (Matched trades matching criteria provided on request).</p> <p>All trades of the specified account made within the specified period are returned. If no account is specified, then all trades of allowed accounts (accounts specified in the Accounts allowed for session field of FIX session settings in TP) are returned.</p>	Y	Int

Tag	Tag name	Description	Required	Data type
263	SubscriptionRequestType	Type of subscription for trade capture reports. Supported value – 0 (Snapshot). A single query for trades.	N	Char
1	Account	FIX-API platform account whose trades to request. If not specified in a TradeCaptureReportRequest message, then trades for all accounts allowed (accounts specified in the Accounts allowed for session field of FIX session settings in TP) for this FIX session are returned.	N	String
580	NoDates	Number of TransactTime(60) fields to set a date range. Supported value – 2 (for start and end of the time period).	Y	NumInGroup
Repeating tags	60	Start (first entry in repeating group) or end (second entry in repeating group) date and time of the period for which trades should be retrieved. Time and date represented in UTC in YYYYMMDD-HH:MM:SSsss format.	Y	UTCTimestamp
StandardTrailer			Y	

TradeCaptureReportRequestAck

TradeCaptureReportRequestAck is a message sent by TP in response to **TradeCaptureReportRequest** message in order to:

- Acknowledge that there are trades within the specified period: **TradeRequestResult(749) = 0**, **TradeRequestStatus(750) = 0**, and **TotNumTradeReports(748)** is not present. This message is followed by consequent **TradeCaptureReport** messages.
- Indicate that no trades matching criteria from the request were found: **TotNumTradeReports(748) = 0**, **TradeRequestResult(749) = 0**, and **TradeRequestStatus(750) = 0**.
- Inform that the trade capture request was invalid, for example, the start date of the requested period was before the end date: **TradeRequestResult(749) = 99** (Other) and **TradeRequestStatus(750) = 2** (Rejected).

Table 7. TradeCaptureReportRequestAck FIX message components

Tag	Tag name	Description	Required	Data type
	StandardHeader		Y	
	MsgType(35) = AQ			
568	TradeRequestID	TradeRequestID(568) of the TradeCaptureReportRequest message for which this TradeCaptureReportRequestAck message is sent.	Y	String
569	TradeRequestType	Type of request. Supported value – 1 (Matched trades matching criteria provided on request).	Y	Int

Tag	Tag name	Description	Required	Data type
263	SubscriptionRequestType	Type of subscription for trade capture report. Supported value – 0 (Snapshot).	N	Char
748	TotNumTradeReports	Total number of TradeCaptureReport messages returned. This tag is only present when no trades were found, in this case, its value is 0 .	N	Int
749	TradeRequestResult	Result of TradeCaptureReportRequest message. <ul style="list-style-type: none"> • 0 – successful (default). • 99 – other. 	Y	Int
750	TradeRequestStatus	Status of trade request: <ul style="list-style-type: none"> • 0 – accepted. • 2 – rejected. 	Y	Int
58	Text	Contains error message when TradeCaptureReportRequest message is rejected.	N	String
<u>StandardTrailer</u>				Y

TradeCaptureReport

TradeCaptureReport is a message sent by TP in response to a **TradeCaptureReportRequest** message after sending a **TradeCaptureReportRequestAck** message.

A separate **TradeCaptureReport** message is sent for each trade (i.e. the corresponding **ExecutionReport** message has **ExecType(150) = F** (Trade)) made within the requested time period.

Table 8. TradeCaptureReport FIX message components

Tag	Tag name	Description	Required	Data type
	StandardHeader		Y	
	MsgType(35) = AE			
568	TradeRequestID	TradeRequestID(568) of TradeCaptureReportRequest message for which this TradeCaptureReport is sent.	N	String
912	LastRptRequested	Indicates whether this report is the last in the sequence of TradeCaptureReport messages: <ul style="list-style-type: none"> • Y – last. • N – not last. 	N	Boolean
17	ExecID	ExecID(17) of the corresponding ExecutionReport message.	N	String

Tag	Tag name	Description	Required	Data type
		<u>Instrument</u>	Y	
32	LastQty	<p>Trade quantity:</p> <ul style="list-style-type: none"> • In base units if T4B Fix-protocol version session setting in TP is FIXAPI 3.2. • In the same units as received in the NewOrderSingle message if T4B Fix-protocol version session setting in TP is FIXAPI 3.1. 	Y	Qty
31	LastPx	Trade price.	Y	Price
75	TradeDate	<p>Date on which the trade occurred. Date of local market is represented in YYYY:MM:DD format.</p>	Y	LocalMktDate
60	TransactTime	<p>Date and time of the transaction represented by this TradeCaptureReport message occurred. Date and time represented in UTC in YYYYMMDD-HH:MM:SS.sss format.</p>	Y	UTCTimestamp
552	NoSides	Number of side entries (BUY/SELL) in the TradeCaptureReport message.	Y	NumInGroup

Tag	Tag name	Description	Required	Data type
		Supported value – 1 (because a TradeCaptureReport message is based on a single ExecutionReport message where only one side is specified).		
Repeating tags	54 Side	<p>Side of an order:</p> <ul style="list-style-type: none"> • 1 – BUY. • 2 – SELL. 	Y	Char
	37 OrderID	<p>Identifier of an order for which trade is provided.</p> <p>It is the OrderID(37) from an ExecutionReport message with ExecType(150) = F (Trade).</p>	Y	String
	11 ClOrdID	<p>Identifier supplied by a FIX client of the order for which trade is provided.</p> <p>It is ClOrdID(11) from a NewOrderSingle message for which the ExecutionReport message is received.</p>	N	String
	1 Account	<p>FIX-API platform account on behalf of which the trade was made.</p> <p>TP always fills in this field.</p>	N	String
StandardTrailer			Y	

RequestForPositions

RequestForPositions is a message sent by a FIX client to request for positions for the specified FIX-API platform account.

In response, TP sends a **PositionReport** message with position information.

Table 9. RequestForPositions FIX message components

Tag	Tag name	Description	Required	Data type
		<u>StandardHeader</u>	Y	
		MsgType(35) = AN		
710	PosReqID	Unique identifier of the RequestForPositions message.	Y	String
724	PosReqType	Type of position request. Supported value – 0 (Positions).	Y	Int
1	Account	FIX-API platform account for which positions are requested.	Y	String
	<u>Instrument</u>	If Instrument is not specified, then positions are requested for all symbols.	N	
	<u>StandardTrailer</u>		Y	

PositionReport

PositionReport is a message sent by TP in response to a **RequestForPositions** message.

Table 10. PositionReport FIX message components

Tag	Tag name	Description	Required	Data type
		<u>StandardHeader</u>	Y	
		MsgType(35) = AP		
710	PosReqID	Unique identifier from the RequestForPositions message.	Y	String
724	PosReqType	Type of position request. Supported value – 0 (Positions).	Y	Int
1	Account	FIX-API platform account for which positions are returned.	Y	String
728	PosReqResult	Result of the position request. Supported values: <ul style="list-style-type: none"> • 0 – valid request. • 1 – invalid or unsupported request. • 2 – no positions found that match criteria. 	Y	Int

Tag	Tag name	Description	Required	Data type
	<u>Instrument</u>	If not specified in the RequestForPositions message, positions are returned for all symbols. In this case, symbols are specified inside the NoPositions(702) group.	N	
702	NoPositions	Number of position entries.	Y	NumInGroup
	703 PosType	Position quantity. Supported value – TOT (Total transaction quantity). Values returned in LongQty(704) and ShortQty(705) represent the total quantity of an open position.	Y	String
Repeating tags	704 LongQty	BUY position volume: <ul style="list-style-type: none"> • In base units if T4B Fix-protocol version session setting in TP is FIXAPI 3.2. • In the same units as received in the NewOrderSingle message if T4B Fix-protocol version session setting in TP is FIXAPI 3.1. 	Y	Qty
	705 ShortQty	SELL position volume: <ul style="list-style-type: none"> • In base units if T4B Fix-protocol version session setting in TP is FIXAPI 3.2. 	Y	Qty

Tag	Tag name	Description	Required	Data type
		<ul style="list-style-type: none"> In the same units as received in the NewOrderSingle message if T4B Fix-protocol version session setting in TP is FIXAPI 3.1. 		
	<u>Instrument</u>	<p>Instrument for which positions are returned.</p> <p>This field is needed to specify the position symbol if no symbol is specified in the RequestForPositions message.</p>	N	
58	Text	<p>Free format text string.</p> <p>May return details on criteria that did not match for PosReqResult(728) = 2.</p>	N	String
	<u>StandardTrailer</u>		Y	

MARKET SESSION MESSAGES

MarketDataRequest

MarketDataRequest is a message sent by a FIX client to subscribe to market data or unsubscribe from an existing market data subscription (**SubscriptionRequestType(263) = 2**).

In response, TP sends an initial set of market data in the **MarketDataSnapshotFullRefresh** message. Further, market data updates are provided in:

- **MarketDataSnapshotFullRefresh** messages if **SubscriptionRequestType(263) = 0**.
- **MarketDataIncrementalRefresh** messages if **SubscriptionRequestType(263) = 1**.

If a **MarketDataRequest** message is rejected, then the **MarketDataRequestReject** message is sent in response.

If a **MarketDataRequest** message is sent for multiple symbols, then, for each symbol, market data is provided in a separate stream of messages with the same **MDReqID(262)** as in the initial **MarketDataRequest** message.

To unsubscribe from market data, a **MarketDataRequest** message should be sent with **SubscriptionRequestType(263) = 2** (Unsubscribe) and **MDReqID(262)** of the earlier **MarketDataRequest** message whose subscription must be disabled (subscription will always be disabled for all symbols specified in the initial subscription message, with no regards to the symbols specified in the unsubscription message).

Table 11. MarketDataRequest FIX message components

Tag	Tag name	Description	Required	Data type
	StandardHeader		Y	
	MsgType(35) = V			
262	MDReqID	<p>Market data request identifier:</p> <ul style="list-style-type: none"> For requests with SubscriptionRequestType(263) = 0 or 1, a unique identifier. For requests with SubscriptionRequestType(263) = 2, it is the MDReqID(262) of the earlier MarketDataRequest message whose subscription must be disabled. 	Y	String
263	SubscriptionRequestType	<p>Type of requested subscription:</p> <ul style="list-style-type: none"> 0 – Snapshot + Updates (updates are provided in MarketDataSnapshotFullRefresh messages). 1 – Snapshot + Updates (updates are provided in MarketDataIncrementalRefresh messages). 2 – Unsubscribe. 	Y	Char
264	MarketDepth	<p>Requested depth of market:</p> <ul style="list-style-type: none"> 0 – full depth (maximum of 20 levels). 	Y	Int

Tag	Tag name	Description	Required	Data type
		<ul style="list-style-type: none"> • 1 – top of book. • 2-20 – top N market depth levels. 		
265	MDUpdateType	<p>Type of market data update.</p> <p>Supported value – 0 (Full refresh).</p> <p>Note: Type of market data update is defined by SubscriptionRequestType(263).</p>	N	Int
267	NoMDEntryTypes	Number of MDEntryType(269) fields requested.	Y	NumInGroup
Repeating tags	269	<p>Type of market data entry:</p> <ul style="list-style-type: none"> • 0 – Bid. • 1 – Offer. <p>Note: TP only allows subscriptions to both Bid (0) and Offer (1) at the same time. If only 0 or 1 is specified in the MarketDataRequest message, then the request is rejected with the MarketDataRequestReject message.</p>	Y	Char
	146	Specifies the number of Instrument components requested.	Y	NumInGroup

Tag	Tag name	Description	Required	Data type
	<u>Instrument</u>		Y	
Repeating tags				
	<u>StandardTrailer</u>		Y	

MarketDataSnapshotFullRefresh

MarketDataSnapshotFullRefresh is a message sent by TP to provide an initial set of market data and also further updates (if **SubscriptionRequestType(263) = 0** in the **MarketDataRequest** message).

TP sends a **MarketDataSnapshotFullRefresh** message with the initial set of market data after a FIX client has subscribed by sending a **MarketDataRequest** message (**SubscriptionRequestType(263) = 0**). Consequent **MarketDataSnapshotFullRefresh** messages are sent automatically to update market data.

If a FIX client subscribed for multiple symbols, then for each symbol, there is a separate stream of **MarketDataSnapshotFullRefresh** messages.

Note: FIX-API platform supplies market data in accordance with its [Feeding configuration](#) in TP.

Table 12. MarketDataSnapshotFullRefresh FIX message components

Tag	Tag name	Description	Required	Data type
	StandardHeader		Y	
	MsgType(35) = W			
262	MDReqID	MDReqID(262) of the MarketDataRequest message with which the subscription was initiated.	N	String
	Instrument		Y	
268	NoMDEntries	Number of entries in a market data message.	Y	NumInGroup

Tag	Tag name	Description	Required	Data type
Repeating tags	269 MDEntryType	Type of the market data entry: <ul style="list-style-type: none"> • 0 – Bid • 1 – Offer 	Y	Char
	270 MDEntryPx	Price of the market data entry.	N	Price
	271 MDEntrySize	Volume of the market data entry in base units, unless T4B Fix-protocol version session setting in TP is set to FIXAPI 3.1 .	N	Qty
	272 MDEntryDate	Date of the market data entry represented in UTC in YYYYMMDD format.	N	UTCDateOnly
	273 MDEntryTime	Time of the market data entry represented in UTC in HH:MM:SS.sss format.	N	UTCTimeOnly
StandardTrailer				Y

MarketDataIncrementalRefresh

MarketDataIncrementalRefresh is a message sent by TP to provide market data incremental updates if **SubscriptionRequestType(263) = 1** in the **MarketDataRequest** message.

TP starts sending **MarketDataIncrementalRefresh** messages after a FIX client has subscribed for market data by sending a **MarketDataRequest** message (**SubscriptionRequestType(263) = 1**) and received the initial set of market data in **MarketDataSnapshotFullRefresh** message. **MarketDataIncrementalRefresh** messages are sent automatically to a FIX client once TP receives updated market data.

A FIX client should store a full set of market data, and when a **MarketDataIncrementalRefresh** message is received, update it based on **MDUpdateAction(279)**:

- **MDUpdateAction(279) = 0** – a new market data entry must be added to the stored set of market data.
- **MDUpdateAction(279) = 1** – an existing market data entry must be changed.
- **MDUpdateAction(279) = 2** – an existing market data entry must be deleted.

If a FIX client subscribed for multiple symbols, then for each symbol, there is a separate stream of **MarketDataIncrementalRefresh** messages.

Note: FIX-API platform supplies market data in accordance with its [Feeding configuration](#) in TP.

Note: A **MarketDataIncrementalRefresh** message does not contain the **MdEntryId** tag to identify a market entry, instead **MDEntryPx(270)** is used to identify an entry within one market data stream. See the **Figure 1** below for an example:

Figure 1. Process of market data update

1. The initial **MarketDataSnapshotFullRefresh** entries:

MDEntryType	MDEntryPx	MDEntrySize
Bid	0.99396	1500000
Bid	0.99394	3500000
Offer	0.99402	2000000

2. The received **MarketDataIncrementalRefresh** entries:

MDEntryType	MDEntryPx	MDEntrySize	MDUpdateAction
Bid	0.99396	0	Delete
Bid	0.99394	0	Delete
Bid	0.99398	3500000	New
Bid	0.99397	500000	New
Offer	0.99402	2500000	Change

3. The updated book entries:

MDEntryType	MDEntryPx	MDEntrySize
Bid	0.99398	3500000
Bid	0.99397	500000
Offer	0.99402	2500000

Table 13. MarketDataIncrementalRefresh FIX message components

Tag	Tag name	Description	Required	Data type
		<u>StandardHeader</u>	Y	
		MsgType(35) = X		
262	MDReqID	MDReqID(262) of the MarketDataRequest message with which the subscription was initiated.	N	String
268	NoMDEntries	Number of entries in a market data message. Note: The number of entries may be different compared to the number in the initial MarketDataSnapshotFullRefresh message (e.g. if some entry is deleted then a new entry is added).	Y	NumInGroup
		Market data update action: <ul style="list-style-type: none">• 0 – New (new entry added).• 1 – Change (existing entry changed).• 2 – Delete (existing entry deleted).	Y	Char
	269	MDEntryType Type market data entry: <ul style="list-style-type: none">• 0 – Bid.• 1 – Offer.	Y	Char

Tag	Tag name	Description	Required	Data type
	Instrument			N
270	MDEntryPx	Price of market data entry. Used to identify a specific market data entry. TP always fills in this field.	N	Price
271	MDEntrySize	Volume of the market data entry in base units, unless T4B Fix-protocol version session setting in TP is set to FIXAPI 3.1 . TP always fills in this field.	N	Qty
272	MDEntryDate	Date of market data entry represented in UTC in YYYYMMDD format.	N	UTCDateOnly
273	MDEntryTime	Time of market data entry represented in UTC in HH:MM:SS.sss format.	N	UTCTimeOnly
StandardTrailer			Y	

MarketDataRequestReject

MarketDataRequestReject is a message sent by TP in response to the **MarketDataRequest** message in cases when this request is rejected.

Table 14. MarketDataRequestReject FIX message components

Tag	Tag name	Description	Required	Data type
		<u>StandardHeader</u> MsgType(35) = Y	Y	
262	MDReqID	MDReqID(262) of the MarketDataRequest message with which the subscription was initiated.	Y	String
281	MdReqRejReason	Reason for market data request rejection: <ul style="list-style-type: none"> • 0 – unknown symbol. • 1 – duplicate MDReqID(262). • 4 – unsupported SubscriptionRequestType(263). • 5 – unsupported MarketDepth(264). • 6 – unsupported MDUpdateType(265). • 8 – unsupported MDEntryType(269). 	N	Char
58	Text	Free format text string.	N	String

Tag	Tag name	Description	Required	Data type
		Can be used to describe the reason for rejection.		
	<u>StandardTrailer</u>		Y	

OTHER MESSAGES

BusinessMessageReject

BusinessMessageReject is a message sent if an application message is structurally valid, but fails business rules (e.g. if some of the conditionally required fields are not present in an incoming FIX message).

Table 15. BusinessMessageReject FIX message components

Tag	Tag name	Description	Required	Data type
	StandardHeader		Y	
	MsgType(35) = j			
45	RefSeqNum	MsgSeqNum(34) of the rejected FIX message.	N	SeqNum
372	RefMsgType	The MsgType(35) of FIX message being referenced.	Y	String
379	BusinessRejectRefID		N	
380	BusinessRejectReason	Code to identify the reason for rejection: • 0 – other.	Y	Int

Tag	Tag name	Description	Required	Data type
		<ul style="list-style-type: none"> ● 1 – unknown ID. ● 2 – unknown security. ● 3 – unknown message type. ● 4 – application not available. ● 5 – conditionally required field missing. ● 6 – not authorised. ● 7 – DeliverTo firm not available. 		
58	Text	<p>Free format text string. Can be used to describe the reason for rejection.</p>	N	String
<u>StandardTrailer</u>				Y



TOOLS
FOR
BROKERS

CONTACT US

Hopefully, you found our documentation clear, precise, and useful.

If you have any questions or suggestions about the product,
do not hesitate to contact our technical support team via:

- 👤 [TFB customer portal](#)
- ✉️ Email support@t4b.com
- ↗️ Telegram @t4b_support
- 💬 Live chat with our specialist from [our website](#)

Our team will also gladly consult you on how you may adjust the product
to your individual requirements.

Stay informed about our latest news, promotions, and updates
by joining our official [WhatsApp](#) and [Telegram](#) channels.

Thank you for your feedback!