



**TRADING
TECHNOLOGIES**

TT FIX ADAPTER REFERENCE MANUAL

VERSION 7.8.X
DOCUMENT VERSION 7.8.0.DV1 3/5/14



LEGAL

This document and all related computer programs, example programs, and all TT source code are the exclusive property of Trading Technologies International, Inc. ("TT"), and are protected by licensing agreements, copyright law and international treaties. Unauthorized possession, reproduction, duplication, or dissemination of this document, or any portion of it, is illegal and may result in severe civil and criminal penalties.

Unauthorized reproduction of any TT software or proprietary information may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Information in this document is subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of TT.

All trademarks displayed in this document are subject to the trademark rights of TT, or are used under agreement by TT. These trademarks include, but are not limited to, service brand names, slogans and logos and emblems including but not limited to: Trading Technologies®, the Trading Technologies Logo, TT™, X_TRADER®, X_RISK®, MD Trader®, Autospreader®, X_STUDY®, TT_TRADER®, TT CVD®, ADL®, Autotrader™, TT Trainer™, Back Office Bridge™, TTNET™. All other referenced companies, individuals and trademarks retain their rights. All trademarks are the property of their respective owners. The unauthorized use of any trademark displayed in this document is strictly prohibited.

Copyright © 2004-2014 Trading Technologies International, Inc.
All rights reserved.



Table of Contents

About this Manual	7
What's New in TT FIX Adapter 7.8.X?	9
Chapter 1: Introduction	13
Overview	14
Development environments	16
Chapter 2: Software Design Considerations	17
Overview	18
Managing the order book state	20
Managing order book state: persistent FIX order sessions	21
Managing order book state: non-persistent FIX order sessions	23
Non-persistent FIX order sessions: using Tag 11 (ClOrdId) and Tag 37 (OrderId)	26
Chapter 3: About FIX Messages	29
About the message structure	30
Message structure: message header	31
Message structure: message body	36
Component block: Instrument (FIX client request)	37
Component block: Instrument (TT FIX Adapter response)	41
Component block: Underlying instrument	44
Component block: Trader	49
Understanding Account Defaults	54
Account Defaults: Order Matching and Routing Examples	57
Message structure: message footer	61
Chapter 4: Session-Level Messages	63
About session-level messages	64
Logon (A)	65
Logout (5)	67
Heartbeat (0)	68
Test Request (1)	69
Resend Request (2)	70
Sequence Reset (4)	71
Session-Level Reject (3)	72
Chapter 5: Application-Level Messages	73
About application-level messages	74
Application message categories	76
Chapter 6: Application-Level Messages: Pre-Trade	77
About Pre-trade Messages	78

Table of Contents

Market Data Request (V)	79
Market Data Request Reject (Y)	82
Market Data Snapshot/Full Refresh (W)	83
Market Data - Incremental Refresh (X)	89
Security Definition Request (c)	96
Security Definition (d)	98
Security Status Request (e)	102
Security Status (f)	103
Chapter 7: Application-Level Messages: Trade-Related.	105
About trade-related messages	106
Execution Report (8)	107
New Order - Single (D)	121
Order Cancel Reject (9)	126
Order Cancel/Replace Request (G)	129
Order Cancel Request (F)	134
Order Status Request (H)	136
Chapter 8: Application-Level Messages: Post-Trade	137
About post-trade messages	138
Request for Position (UAN)	139
Position Report (UAP)	141
Chapter 9: Application-Level Messages: Other.	155
About other messages	156
Business Message Reject (j)	157
Gateway Status Request (UAR)	158
Gateway Status (UAT)	159
Chapter 10: FIX Message Conversations	161
Logon and logout	162
Security download	165
Security status	171
Market data	173
New order	178
Change order	186
Positions and fills	190
Order status	196
Gateway status	198
Staged Orders	199
Appendix A: Enumerations and Codes	205
Enumerations	206
Price display codes	209
Appendix B: Supported FIX Messages and Tags	211
FIX messages	212
FIX tags by name	214
FIX tags by number	219

Appendix C: P&L Calculation Algorithm	225
Appendix D: TT Gateway-Specific Notes	229
BrokerTec Notes	230
CBOE Notes	231
Eurex Notes	232
NYSE_Liffe Notes	233
ICE Notes	234
TFX Notes	235
OM Gateway Notes	236
LME Gateway Notes	237

About this Manual

Overview

This manual describes the TT FIX Adapter and how it uses FIX tags to communicate with third-party client applications.

This manual includes content that was accurate when published. However, for the most up-to-date information, please refer to TT's new online Help Library.

The Help Library provides access to online versions of TT manuals across all products. The library is located at <https://www.tradingtechnologies.com/en/support/help-library>.

Note: TT no longer provides an updated .pdf version of this manual for future versions of TT FIX Adapter. The online Help Library is the only source of help documentation. Users without Internet access can request a .pdf version of this manual via email at documentation.department@tradingtechnologies.com or by contacting your Account Manager.

Assumptions

This manual assumes the following:

- The target audience consists of FIX Developers with a knowledge of FIX protocol.
- The user has an intermediate or greater level of technical proficiency.
- The user has little or no training in Trading Technologies software.

Required tags

The manual uses the following terminology as to whether particular messages require certain tags. The **Required** column in each message type table uses the following values:

- **Y:** The FIX message must include this tag.
- **C:** If a particular condition is met, then the FIX message must include this tag.
- **N:** Although the FIX message does not require this tag, you can use it to provide any additional benefit as needed.

Conventions

This guide uses a set of terms, symbols, and typographic conventions to categorize specific information. Familiarity with these conventions helps you use this documentation more effectively.

Convention	Use
Bold	Indicates command buttons, menu options, folder/directory names, and file names. Also used for FIX message names.
<i>Italic</i>	Indicates variables, including variable text. Variable text is used when dialog boxes or their components are not labeled. Variable text is also used for labels that change dynamically based on their current context. The wording of variable text does not exactly match what you see on your screen.

Convention	Use
Monospace	Sample code, arguments, properties, arguments, methods, events. Text in Courier New is also 1 point size smaller than the surrounding text.
Monospace bold	FIX tag names or anything that you must type exactly as it appears. Text in Courier New is also 1 point size smaller than the surrounding text.
*	Indicates a wildcard that signifies any number of variable characters. For instance, in the case of *.doc, the asterisk is a wildcard, and *.doc signifies any file ending in ".doc".
[argument]	Optional argument.
‡	Separator character used in FIX messages.
Blue text	Highlighted FIX tags. FIX message conversations in Chapter 10, FIX Message Conversations , on page 161, highlight specific tags relevant for the specific use case.
<u>Underlined blue text</u>	Hyperlink to a web site.

Related documentation

This manual is part of a comprehensive Trading Technologies document set. For additional manuals and documentation on other TT™ topics, contact your TT™ sales representative.

For more information about the FIX 4.2 specification, visit the www.fixprotocol.org web site.

What's New in TT FIX Adapter 7.8.X?

Version 7.8.6 enhancements	FIX Adapter 7.8.6 introduced the following enhancements: <ul style="list-style-type: none">• Support for the Eris_GovEx Gateway
Version 7.8.5 enhancements	FIX Adapter 7.8.5 introduced the following enhancements: <ul style="list-style-type: none">• Added C (Expired) as a new value for Tag 39 (OrdStatus) and Tag 150 (ExecType) in Execution Report (8) messages to support the new FACT Send Expired Order Statuses configuration option, which allows Order Routing and Drop Copy FIX Adapters to send expired order statuses instead of canceled order statuses when a TT Gateway provides an expired order status. This feature requires TT Gateway versions 7.16.9 or later.• Added a Send Unsolicited Rejects FACT configuration option that determines whether Order Routing and Drop Copy FIX Adapters send Execution Report (8) messages for unsolicited rejects.• Added support for the product-level price reasonability checks that can be configured in TT User Setup 7.4.11 and above.
Version 7.8 enhancements	FIX Adapter 7.8 introduced the following enhancements: <ul style="list-style-type: none">• Added support for the new Prevent orders that are more than +/- X ticks away from the market price setting for the FIX client user in TT User Setup. This only applies to FIX clients of Order Routing FIX Adapters.• Added support for the new GTC/GTDate orders allowed setting for the FIX client user in TT User Setup. This only applies to FIX clients of Order Routing FIX Adapters.• Added support for submitting staged orders through the TT Synthetic Strategy Engine (SSE) Gateway• Updated requirements to note support only for TT CME Gateway 7.13 or higher• Added the following tags to the New Order Single (D) and Order Cancel Replace Request (G) messages for staged order support:<ul style="list-style-type: none">- Tag 21 (HandlInst)- Tag 16104 (TTUserTagData)- Tag 16105 (TTOrderTagData)- Tag 16106 (StagedOrderMsg)- Tag 16111 (RoutingLevel)• Added the following tags to the Execution Report (8) and Position Report (UAP) messages to support user-supplied data:<ul style="list-style-type: none">- Tag 16104 (TTUserTagData)- Tag 16105 (TTOrderTagData)• Added the following tags to the Execution Report (8) messages to support staged orders:<ul style="list-style-type: none">- Tag 21 (HandlInst)- Tag 16106 (StagedOrderMsg)- Tag 16109 (StagedOrderStatus)- Tag 16110 (StagedOrderOwner)

- Tag 16111 (**RoutingLevel**)
- Tag 18208 (**TTStrategyEngine**)
- Added the following tags to the **Position Report (UAP)** messages to support synthetic orders:
 - Tag 21 (**HandlInst**)
 - Tag 18208 (**TTStrategyEngine**)
- Added the NoLinks repeating group to the **Execution Report (8)** and **Position Report (UAP)** messages. This group exposes Link IDs and LinkType data on child orders and fills from synthetic orders. It includes the following tags:
 - Tag 16112 (**NoLinks**)
 - Tag 16113 (**LinkID**)
 - Tag 16114 (**LinkType**)
- Added Tag 76 (**ExecBroker**) to the **Execution Report (8)** and **Position Report (UAP)** messages to give up and clearing member information
- Added support for indicative market data by adding the following values to Tag 269 (**MDEntryType**):
 - **r**: indicative bid
 - **s**: indicative ask
 - **t**: indicative settle
- Added the following tags to the **Security Definition (d)** message to retrieve security descriptions and aliases from TT Guardian:
 - Tag 107 (**SecurityDesc**)
 - Tag 18207 (**TTSecurityAlias**)
- Added Tag 18216 (**ExchCred**) to the **Execution Report (8)** and **Position Report (UAP)** messages for orders and fills to provide the exchange credential that was used to route the order, when provided by the TT Gateway.
- Added Tag 19 (**ExecRefID**) to support fill corrections per the FIX 4.2 specification.
- Added Tag 10828 (**TrdType**) to the **Execution Report (8)** and **Position Report (UAP)** messages to distinguish wholesale trades.
- Added Tag 20 (**ExecTransType**) to **Position Report (UAP)** messages.
- Added support for Best Limit if Touched (BLIT) orders by adding a new value "**X**: Best Limit if Touched (BLIT)" to Tag 40 (**OrdType**).
- Added support for the L (Balance of Week), X (Custom), A (Same Day), N (Next Day), and B (Balance of Month) contract terms by adding the following values to Tag 18211 (**ContractTerm**):
 - **A**: (Same Day)
 - **B**: (Balance of Month)
 - **L**: (Balance of Week)
 - **N**: (Next Day)
 - **X**: (Custom)
- Added a new value "**3**: Unsupported Message Type" to Tag 380 (**BusinessRejectReason**).
- Updated Tag 207 (**SecurityExchange**) to be conditional. It is required if Tag 48 (**SecurityID**) is sent.
- Added Tag 16017 (**OrigExecID**) to the **Execution Report (8)** and

Position Report (UAP) messages for fill corrections.

- Added Tag 16018 (**TTSessionID**) to the **Execution Report (8)** and **Position Report (UAP)** messages for fills to provide the TT Gateway's Session ID, when provided by the TT Gateway.
- Added support for the new **Prevent orders that are more than +/- X ticks away from the market price** setting for the FIX client user in TT User Setup. This only applies to FIX clients of Order Routing FIX Adapters.
- Added support for the new **GTC Orders Allowed** setting for the FIX client user in TT User Setup. This only applies to FIX clients of Order Routing FIX Adapters.

Version 7.8 removed features

Made changes in the FIX Adapter Configuration Tool (FACT) to remove the options to send or not send the following tags. TT FIX Adapter now always sends them if available, so FIX clients must handle the tags.

- Tag 15 (**Currency**)
- Tag 318 (**UnderlyingCurrency**)
- Tag 440 (**ClearingAccount**)
- Tag 864 (**NoEvents**)
- Tag 865 (**EventType**)
- Tag 866 (**EventDate**)
- Tag 6038 (**OrderEnteredTime**)
- Tag 10527 (**SecondaryExecID**)
- Tag 10553 (**TTUsername**)
- Tag 16102 (**FFT2**)
- Tag 16103 (**FFT3**)
- Tag 18203 (**ExchangeGateway**)
- Tag 18210 (**PriceFeedStatus**)

For more information about FACT changes, refer to the *TT FIX Adapter System Administration Manual*.

Chapter overview This chapter provides an overview of the TT FIX Adapter.

In this chapter

Section	Page
Overview	14
Development environments	16

Overview

System architecture

TT FIX Adapter enables traders with FIX-based client applications to connect to various exchanges via TT Gateways. As a developer, you must develop the connection from the FIX-based client to the TT FIX Adapter.

TT supports a subset of the FIX protocol (Version 4.2 with Errata 20010501) message types and tags within the message types. Refer to the website <http://www.fixprotocol.org> for additional FIX tag descriptions and message definitions.

Note: TT FIX Adapter supports only those messages and tags described in this document. Do not submit other tags to TT FIX Adapter, as doing so can produce unexpected results.

Supported TT Gateways

TT FIX Adapter supports the following TT Gateways:

- BrokerTec, version 7.13.0 or higher
- BVMF, version 7.15 or higher
- CBOE, version 7.13.0 or higher
- CBOT, version 7.13 or higher
- CFE, version 7.16 or higher
- CME, version 7.13 or higher
- EBS, version 7.13.0 or higher
- eCBOT, version 7.13.0 or higher
- Eris_GovEx, version or 7.16.9 higher
- Eurex, version 7.13.0 or higher
- ICE, version 7.13.0 or higher
- LME, version or higher
- MEFF, version 7.13.0 or higher
- MX, version 7.16 or higher
- Montreal, version 7.13.0 or higher
- NYSE_Liffe, version 7.15 or higher
- NYSE_Liffe_US, version 7.16 or higher
- OSE, version 7.14 or higher
- SFE, version 7.13.0 or higher
- SGX, version 7.13.0 or higher
- TFX, version 7.14 or higher
- TT Synthetic Strategy Engine (SSE), version 7.2.10 or higher
- TSE, version 7.15 or higher
- TOCOM, version 7.13.0 or higher
- TTSIM, version 7.2 or higher

Some exchanges might list some products on other exchanges. For example, the European Climate Exchange lists products on ICE_IPE. For those products, you must specify **ICE_IPE** in Tag 207 (**SecurityExchange**). For information about how to map these exchanges, refer to the section called [Tag 207 \(SecurityExchange\)](#), on page 206.

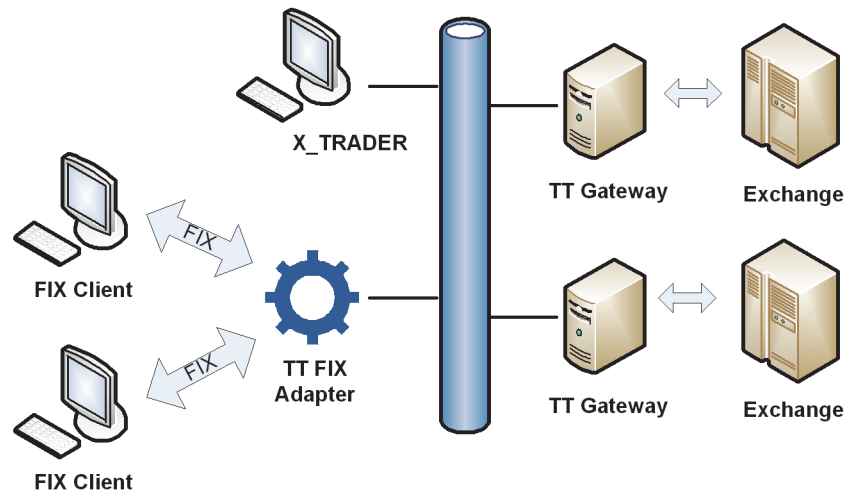
Note: Connectivity to TT Synthetic Strategy Engine (SSE) is only supported for staged orders. Additionally, the FIX Adapter Server user's Gateway Logins must be the same or a subset of those used by the TT Synthetic Strategy Engine (SSE).

Note: The MEFF Gateway does not fully support MEFF options contracts. As a result, TT FIX Adapter does not support MEFF options contracts.

Note: The TT Xetra Gateway supports trading activity via X_TRADER only. Customers are advised that serious issues can occur when trading on the TT Xetra Gateway via FIX Adapter, or when connecting a Drop Copy FIX Adapter to a TT Xetra Gateway.

Order visibility within the TT system

The following illustration shows a typical TT network in which traders use FIX clients and X_TRADER to route orders.



TT FIX Adapter maps all order sessions to a set of TT Member/Group/Trader (MGT) credentials. Thus, all orders routed by FIX clients through FIX Adapter will be visible in other TT applications (such as X_TRADER, X_RISK, etc...) provided that their credentials (MGT) have permission to view the order. Conversely, all orders routed through other TT applications will be visible to FIX clients provided that their credentials (MGT) have permission to view the order.

Development environments

Types of development environments

In order of most recommended to least recommended, TT recommends that you write to the TT FIX Adapter using one of the following environments:

- Exchange simulation
- TT simulation
- Production

Table 1 describes these environments.

Environment	Description
Exchange simulation	In a simulation environment, the TT Gateway connects to the exchange's simulation environment. Because you have access to the exchange's actual behavior and complete safety from a production environment, TT recommends using this environment to develop your TT FIX client.
TT simulation	If you cannot access the exchange's simulation environment, TT recommends that you connect the TT FIX Adapter to a TTSIM Gateway instead.
Production	In a production environment, the TT Gateway connects to the exchange's production environment. Because of the direct connection into the production environment and dangers associated with such a connection, TT discourages development in this sort of environment.

Table 1. TT FIX Adapter development environments

For information about setting up a TT Gateway in Production or Simulation mode, refer to the *TT Gateway System Administration Manual* and the SAM specific to the TT Gateway you want to deploy.

For information on setting up a TTSIM Gateway, refer to the *TT Simulation and Training Administration Manual*.

QuickFIX FIX 4.2 Data Dictionary

QuickFIX is a free and open source implementation of the FIX protocol. It is located at <http://www.quickfixengine.org/>. The QuickFIX distribution contains a default XML data dictionary for each version of FIX used to validate FIX messages, named **FIX40.xml**, **FIX41.xml**, **FIX42.xml**, etc.

The QuickFIX Data Dictionary for TT FIX Adapter is a tailored version of the QuickFIX FIX 4.2 default data dictionary ("FIX42.xml"). It is intended to provide those users of the QuickFIX FIX Engine who want to connect to TT FIX Adapter with a complete set of the required validation rules, thereby minimizing project initiation time.

TT stores the QuickFIX Data Dictionary in the TT FIX Adapter distribution in **<root_drive>:\tt\fixadapter\FIX42.xml**.

2

Software Design Considerations

Chapter overview

This chapter contains important information about designing software that communicates with TT FIX Adapter.

Note: TT strongly recommends that all FIX client developers read this chapter thoroughly before starting the development process.

In this chapter

Section	Page
Overview	18
Managing the order book state	20
Managing order book state: persistent FIX order sessions	21
Managing order book state: non-persistent FIX order sessions	23
Non-persistent FIX order sessions: using Tag 11 (ClOrdId) and Tag 37 (OrderId)	26

Overview

Price vs. order FIX sessions

Table 2 lists the types of FIX sessions that TT FIX Adapter supports. A single FIX client can create one or more of each session type, based on its processing needs.

Session type	Uses
Price	<ul style="list-style-type: none"> • Subscribe for market data • Subscribe for TT Gateway status updates • Subscribe for security definitions • Subscribe for security status updates
Order	<ul style="list-style-type: none"> • Submit orders • Receive order acknowledgements and fills • Subscribe for TT Gateway status updates • Subscribe for security definitions

Table 2. Comparison of price and order FIX sessions

Supported FIX messages

In addition to different types of FIX sessions, the TT FIX Adapter also supports two basic types of configurations:

- Drop copy configurations, which send market data, order acknowledgements and fills to FIX clients
- Order routing configurations, which allows FIX clients to request market data, submit orders, and receive notifications for orders and fills

For more information about TT FIX Adapter configurations, refer to the *TT FIX Adapter System Administration Manual*.

Table 3 shows which messages different types of FIX clients can exchange with TT FIX Adapter. It also shows how the session type and session persistence affect which messages they can exchange.

Use case Session type	Drop Copy		Order Routing	
	Price	Order	Price	Order
Heartbeat (0)	Yes	Yes	Yes	Yes
Test Request (1)	Yes	Yes	Yes	Yes
Resend Request (2)	Yes	Yes	Yes	Yes
Session-Level Reject (3)	Yes	Yes	Yes	Yes
Sequence Reset (4)	Yes	Yes	Yes	Yes
Logout (5)	Yes	Yes	Yes	Yes
Logon (A)	Yes	Yes	Yes	Yes
Business Message Reject (j)	Yes	Yes	Yes	Yes
Execution Report (8)	No	Yes	No	Yes
Gateway Status Request (UAR)	Yes	Yes	Yes	Yes
Gateway Status (UAT)	Yes	Yes	Yes	Yes
Market Data Request (V)	Yes	No	Yes	No
Market Data Request Reject (Y)	Yes	No	Yes	No
Market Data Snapshot/Full Refresh (W)	Yes	No	Yes	No
Market Data - Incremental Refresh (X)	Yes	No	Yes	No
New Order - Single (D)	No	No	No	Yes
Order Cancel Request (F)	No	No	No	Yes

Table 3. Application-level message mappings

Use case Session type	Drop Copy		Order Routing	
	Price	Order	Price	Order
Order Cancel/Replace Request (G)	No	No	No	Yes
Order Status Request (H)	No	Yes	No	Yes
Order Cancel Reject (9)	No	No	No	Yes
Request for Position (UAN)	No	Yes	No	Yes
Position Report (UAP)	No	Yes	No	Yes
Security Definition Request (c)	Yes	Yes	Yes	Yes
Security Definition (d)	Yes	Yes	Yes	Yes
Security Status Request (e)	Yes	No	Yes	No
Security Status (f)	Yes	No	Yes	No

Table 3. Application-level message mappings

Managing the order book state

About order book state	<p>In this manual, the term, order book state, refers to the combined status of the following at a moment in time:</p> <ul style="list-style-type: none">• Trader's working orders• Trader's current position• List of fills that constitute a trader's position
-------------------------------	--

Order book state models	<p>TT FIX Adapter offers two models for managing the order book state for FIX order sessions:</p> <ul style="list-style-type: none">• Persistent• Non-persistent
--------------------------------	---

Note: TT FIX Adapter requires all FIX price sessions to be non-persistent.

The following sections explain the differences between using persistent and non-persistent FIX order sessions and describe the considerations you must take into account when developing a FIX client so it correctly manages order book states.

Managing order book state: persistent FIX order sessions

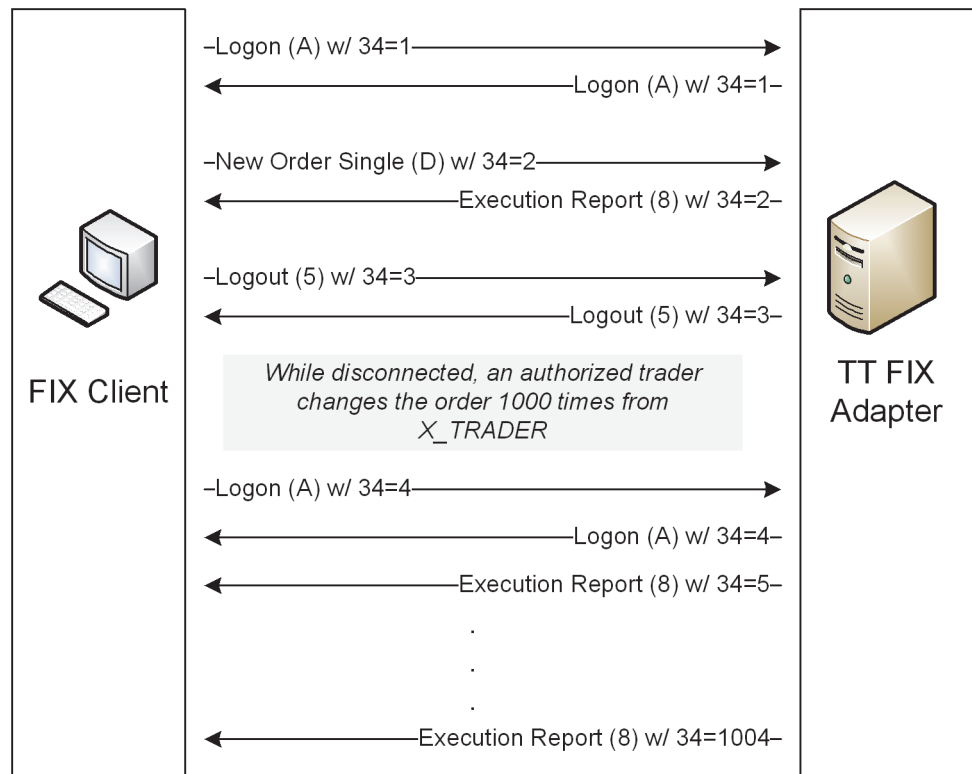
Determining the order book state

Persistent FIX order sessions are stateful. Specifically, FIX clients persist the order book state as well as the last message sequence number received from the TT FIX Adapter. The TT FIX Adapter persists the same information. Therefore, the order book state and message sequence number are linked.

When a FIX client initializes, it should use the following process to determine the current order book state:

- 1 Read its local cache to determine the:
 - Order book state since the last time it connected to the TT FIX Adapter
 - Last message sequence number received from the TT FIX Adapter
- 2 Send a **Logon (A)** message to the TT FIX Adapter with Tag 34 (**MsgSeqNum**)=(1 + <last message sequence number sent to the TT FIX Adapter>).
- 3 Process all order update messages pending from the TT FIX Adapter.

The following example shows how a FIX client using a persistent FIX order session maintains an up-to-date order book, even when it disconnects and reconnects to the TT FIX Adapter.



Note: FIX clients can use either Tag 11 (**clOrdID**) or Tag 37 (**orderID**) when referencing order handling messages, because the TT FIX Adapter preserves both across connections.

Handling missed messages

Persistent FIX sessions support the standard message replay mechanism specified by the FIX Protocol. A TT FIX Adapter honors **Resend Request (2)** messages received from FIX clients because the TT FIX Adapter caches all messages that it sends.

Handling connection problems

During the course of normal operation, various problems, such as hardware failures, can occur. To account for the possibility of failures, TT recommends that you make two duplicate TT FIX Adapters (primary and secondary) available to FIX clients. These redundant adapters should be configured identically so that FIX clients can interact with both in the same manner.

If a FIX client that uses a persistent FIX session cannot communicate with its primary TT FIX Adapter, it should automatically connect to the secondary TT FIX Adapter, reset the session, and determine the current order book state. Specifically, the FIX client should:

- 1 Send a **Logon (A)** message to the secondary TT FIX Adapter with Tag 34 (**MsgSeqNum**)=1 and Tag 141 (**ResetSeqNumFlag**)=Y.
- 2 Request the current state of all orders by sending an **Order Status Request (H)** message, omitting both Tag 11 (**ClOrdID**) and Tag 37 (**OrderID**).
- 3 Determine the trader's current position by using one of the following methods:
 - Requesting all trades, manual fills, SODs, and DSODs since the last rollover by sending sequential **Request For Position (UAN)** messages with:
 - o Tag 16724 (**PosReqType**)=1 (trades)
 - o Tag 16724 (**PosReqType**)=4 (SOD)
 - o Tag 16724 (**PosReqType**)=5 (manual fills)
 - o Tag 16724 (**PosReqType**)=6 (DSOD)

Reconcile the information resulting from these requests with the information that the FIX client already received to determine whether anything was missed up to this point.
 - Request a list of positions based on all trades, manual fills, SODs, and DSODs that occurred since the last rollover by sending a **Request For Position (UAN)** message with Tag 16724 (**PosReqType**)=0 (positions).

Because the TT FIX Adapter maintains the session cache locally for each adapter, FIX clients must log into the secondary TT FIX Adapter with Tag 34 (**MsgSeqNum**)=1 and Tag 141 (**ResetSeqNumFlag**)=Y to begin a new session. Additionally, TT FIX Adapter preserves neither Tag 11 (**ClOrdID**) nor any information regarding how the order book got to its current state. You must use Tag 37 (**OrderID**) to reference all existing orders in all subsequent FIX messages. However, the FIX client can resume normal operations after it resynchronizes its order book.

Managing order book state: non-persistent FIX order sessions

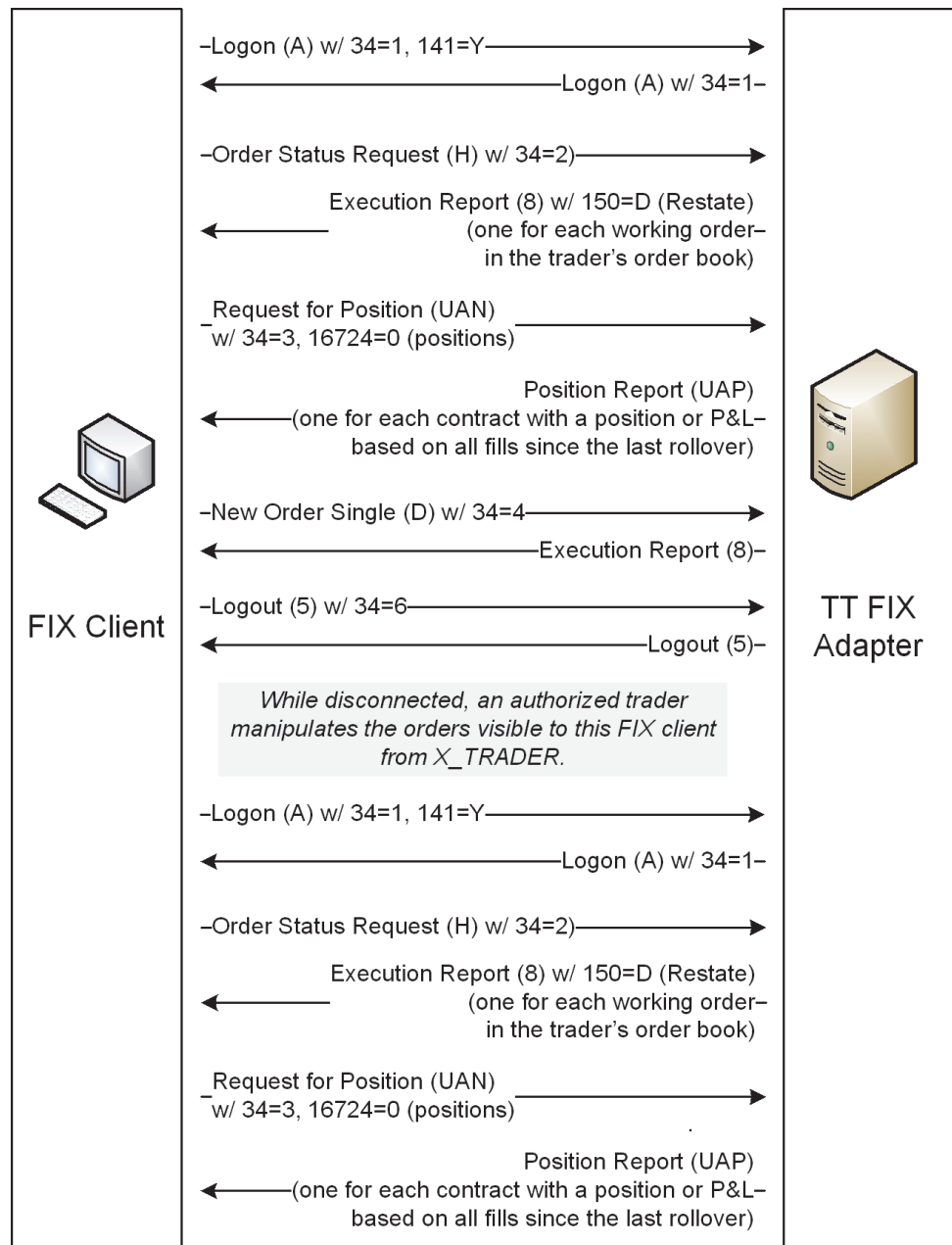
Determining the order book state

During startup, the TT FIX Adapter queries the TT Gateways to determine the order book state of each FIX client. Then, when a FIX client initializes its connection, it queries the TT FIX Adapter for its order book state information. When using this model, the order book state and message sequence numbers stay synchronized only during a session, but not across sessions.

When a non-persistent FIX client initializes, it should use the following process to determine the current order book state:

- 1 Send a **Logon (A)** message to the TT FIX Adapter with Tag 34 (**MsgSeqNum**)=1 and Tag 141 (**ResetSeqNumFlag**)=Y.
- 2 Request the current state of all orders by sending an **Order Status Request (H)** message, omitting both Tag 11 (**ClOrdID**) and Tag 37 (**OrderID**).
- 3 Determine the trader's positions by sending a **Request For Position (UAN)** message with Tag 16724 (**PosReqType**)=0 (positions).
- 4 Optionally request all fills, manual fills, SODs, and DSODs that comprise this position by sending sequential **Request For Position (UAN)** messages with:
 - Tag 16724 (**PosReqType**)=1 (trades)
 - Tag 16724 (**PosReqType**)=4 (SOD)
 - Tag 16724 (**PosReqType**)=5 (manual fills)
 - Tag 16724 (**PosReqType**)=6 (DSOD)

The following example illustrates the exchange of messages between a FIX client and a TT FIX Adapter for non-persistent FIX order sessions. Every time the FIX client establishes a connection, it needs to start a new session (34=1, 141=Y) and request the current state of all working orders. Any changes that occurred while the FIX client was not connected are lost.



Note: After downloading orders that include Tag 37 (**OrderID**), traders can modify orders as long as all change order requests include Tag 37 (**OrderID**). For more information refer to the following section, [Non-persistent FIX order sessions: using Tag 11 \(ClOrdId\) and Tag 37 \(OrderId\)](#).

Handling missed messages

Non-persistent FIX order sessions support the standard message replay mechanism specified by the FIX Protocol. A TT FIX Adapter honors **Resend Request (2)** messages received from FIX clients because the TT FIX Adapter caches all messages that it sends on the order session.

Handling connection problems

During the course of normal operation, various problems, such as hardware failures, can occur. To account for the possibility of failures, TT recommends that you make two redundant TT FIX Adapters (primary and secondary) available to FIX clients. These redundant adapters should be configured identically so that FIX clients can interact with both in the same manner.

FIX clients that use non-persistent FIX sessions should automatically connect to its secondary TT FIX Adapter if it cannot connect to its primary TT FIX Adapter. Because both TT FIX Adapters deliver the same order book information, FIX clients do not need additional coding to handle this failover situation.

Non-persistent FIX order sessions: using Tag 11 (ClOrdId) and Tag 37 (OrderId)

Overview	Because the TT methodology for tags 11 (ClOrdID), 41 (OrigClOrdID), and 37 (OrderID) deviates from the FIX specification, you should use them carefully. For non-persistent sessions, TT FIX Adapter erases all Tag 11 (ClOrdID)s associated with the session from TT FIX Adapter's memory when a session disconnects from TT FIX Adapter, so FIX clients must use Tag 37 (OrderID) in all subsequent Order Cancel Request (F) and Order Cancel Replace Request (G) requests.
Scenario 1: changing an order	<p>The following illustrates how a FIX client and a TT FIX Adapter exchange messages when changing an order in normal operating conditions.</p> <ol style="list-style-type: none"> Send a new order - FIX client sends New Order Single (D) message with: <pre>ClOrdId(11)=TempClOrdId OrderId(37)=<not sent></pre> Order is accepted - TT FIX Adapter sends an Execution Report (8) with: <pre>ExecTransType(20)=0 ExecType(150)=0 ClOrdId(11)=TempClOrdId OrderId(37)=TTOrderId OrigClOrdId(41)=<not sent></pre> Send a cancel/replace - FIX client sends Order Cancel Replace Request (G) message with: <pre>ClOrdId(11)=TempClOrdId2 OrderId(37)=TTOrderId OrigClOrdId(41)=<not sent></pre> <ul style="list-style-type: none"> If Cancel/Replace is accepted, TT FIX Adapter sends an Execution Report (8) with: <pre>ClOrdId(11)=TempClOrdId2 OrderId(37)=TTOrderId OrigClOrdId(41)=<not sent></pre> If Cancel/Replace is rejected by TT FIX Adapter, TT FIX Adapter sends an Order Cancel Reject (9) with: <pre>ClOrdId(11)=TempClOrdId2 OrderId(37)=TTOrderId OrigClOrdId(41)=<not sent></pre> If Cancel/Replace is rejected by TT Gateway or exchange, TT FIX Adapter sends an Order Cancel Reject (9) with: <pre>ClOrdId(11)=TempClOrdId2 OrderId(37)=TTOrderId OrigClOrdId(41)=TempClOrdId</pre>

Scenario 2: modifying an order after a lost connection

The following illustrates how a FIX client and a TT FIX Adapter exchange messages after their connection is interrupted.

- 1 Send a new order - FIX client sends **New Order Single (D)** message with:

```
ClOrdId(11)=TempClOrdId
OrderId(37)=<not sent>
```

- 2 Order is accepted - TT FIX Adapter sends an **Execution Report (8)** with:

```
ExecTransType(20)=0
ExecType(150)=0
ClOrdId(11)=TempClOrdId
OrderId(37)=TTOrderId
OrigClOrdId(41)=<not sent>
```

- 3 The connection between TT FIX Adapter and client is lost. At this point, all Tag 11 (**ClOrdID**) information that was associated with orders is lost because non-persistent sessions automatically reset client order IDs after a reset. However, clients can still use Tag 37 (**OrderID**) to manipulate orders.
- 4 After reconnecting, send a cancel/replace - FIX client sends **Order Cancel Replace Request (G)** message with:

```
ClOrdId(11)=TempClOrdId2
OrderId(37)=TTOrderId
OrigClOrdId(41)=<not sent>
```

- If Cancel/Replace is accepted, TT FIX Adapter sends an **Execution Report (8)** with:

```
ClOrdId(11)=TempClOrdId2
OrderId(37)=TTOrderId
OrigClOrdId(41)=<not sent>
```

- If Cancel/Replace is rejected by TT FIX Adapter, TT FIX Adapter sends an **Order Cancel Reject (9)** with:

```
ClOrdId(11)=TempClOrdId2
OrderId(37)=TTOrderId
OrigClOrdId(41)=<not sent>
```

- If Cancel/Replace is rejected by TT Gateway or exchange, TT FIX Adapter sends an **Order Cancel Reject (9)** with:

```
ClOrdId(11)=TempClOrdId2
OrderId(37)=TTOrderId
OrigClOrdId(41)=<not sent>
```

Scenario 3: modifying an order through X_TRADER

An order is sent, cancel/replaced, or cancelled by X_TRADER using the same MGT and the acknowledgment is forwarded to the FIX client, so TT FIX Adapter sends an **Execution Report (8)** with:

```
ExecTransType(20)=0  
Exec (150)= D (Restate)  
ClOrdId(11)=<not sent>  
OrdId(37)=TTOrderId  
OrigClOrdId(41)=<not sent>
```

TT FIX Adapter sends **Execution Report (8)** with Tag 150 (**ExecType**)=**D** when orders change outside of TT FIX Adapter.

For example, in X_TRADER, the **Change** button is the equivalent of a TT FIX Adapter **Order Cancel Replace Request (G)** message. The X_TRADER **Cancel/Replace** button actually cancels the order on the exchange and then resubmits the order with its changes as a new order. If a trader uses the X_TRADER button, TT FIX Adapter sends an **Execution Report (8)** with Tag 39 (**ordStatus**)=**5** and Tag 150 (**ExecType**)=**D**. Tag 37 (**orderId**) always keeps the same value.

Chapter overview

This chapter describes the basic structure and parts of a FIX message. It also lists the messages that TT FIX Adapter supports for the different types of FIX client applications.

In this chapter

Section	Page
About the message structure	30
Message structure: message header	31
Message structure: message body	36
Component block: Instrument (FIX client request)	37
Component block: Instrument (TT FIX Adapter response)	41
Component block: Underlying instrument	44
Component block: Trader	49
Understanding Account Defaults	54
Account Defaults: Order Matching and Routing Examples	57
Message structure: message footer	61

About the message structure

Overview	FIX messages contain data relevant to the data packet, message, and machine.
Parts of a message	<p>Every message sent between a FIX client and TT FIX Adapter includes the following parts:</p> <ul style="list-style-type: none">• Message header, which contains meta-data about the message• Message body, which contains information specific to the request or response• Message trailer, which terminates every FIX message

Message structure: message header

Overview Every message sent to and received from the TT FIX Adapter must start with message sender and receiver information, message type, and message size.

Message header tags in requests from a FIX client to TT FIX Adapter

Tag #	Field name	Req'd	Comments
8	BeginString	Y	FIX protocol version. The tag indicates the beginning of a new message. This tag must be the first tag in the message. You must set the value to FIX.4.2 . Data type: String
9	BodyLength	Y	Message length (in characters), which represents number of characters in the message following this tag up to, and including, the delimiter immediately preceding Tag 10 (Checksum). This tag must be the second field in a message. Data type: Length
35	MsgType	Y	Specifies the type of message contained in the message body. This tag must appear third in the list of header tags. Data type: String
49	SenderCompID	Y	FIX client TT Universal Login ID Maximum supported length = 11 characters Data type: String
56	TargetCompID	Y	TT FIX Adapter ID, specified as LocalCompID in the TT FIX Adapter configuration. Maximum supported length = 255 characters Data type: String For more information, see the <i>TT FIX Adapter System Administration Manual</i> .
34	MsgSeqNum	Y	Message sequence number. Data type: SeqNum
43	PossDupFlag	C	Whether the sequence number for this message is already used. Possible values include: Y : Possible duplicate N : Original transmission Data type: Boolean Condition: Must send when a FIX client resends messages Note: TT FIX Adapter rejects all New Order Single (D) and Order Cancel Replace Request (G) messages that contain 43=Y.

Table 5. Message header tags in requests from a FIX client to TT FIX Adapter

Tag #	Field name	Req'd	Comments
50	SenderSubID	N	<p>Unique Trader ID (can also be provided in Tag 116 (OnBehalfOfSubID)).</p> <p>Data type: String, but cannot contain a forward slash (/) character</p> <p>For more information about the interaction between tags 50, 57 and 116, see the section called Requests and responses involving Tag 50 (SenderSubID), Tag 57 (TargetSubID), Tag 116 (OnBehalfOfID), and Tag 129 (DeliverToSubID), on page 34.</p>
57	TargetSubID	N	<p>User data.</p> <p>If you submit Tag 57 (TargetSubID), TT FIX Adapter returns the value in Tag 50 (SenderSubID) of the response. If you omit this tag, TT FIX Adapter returns the sender M/G/T in Tag 50 (SenderSubID) of the response.</p> <p>Data type: String, but cannot contain a forward slash (/) character</p> <p>For more information about the interaction between tags 50, 57 and 116, see the section called Requests and responses involving Tag 50 (SenderSubID), Tag 57 (TargetSubID), Tag 116 (OnBehalfOfID), and Tag 129 (DeliverToSubID), on page 34.</p>
116	OnBehalfOfSubID	N	<p>Unique Trader ID (can also be provided in Tag 50 (SenderSubID)).</p> <p>Data type: String, but cannot contain a forward slash (/) character</p> <p>For more information about the interaction between tags 50, 57 and 116, see the section called Requests and responses involving Tag 50 (SenderSubID), Tag 57 (TargetSubID), Tag 116 (OnBehalfOfID), and Tag 129 (DeliverToSubID), on page 34.</p>
122	OrigSendingTime	C	<p>Original time of message transmission, when transmitting orders as the result of a resend request. Always expressed in UTC.</p> <p>Data type: UTCTimestamp</p> <p>Condition: Required for resent messages</p>
52	SendingTime	Y	<p>Time, in UTC, the message was sent.</p> <p>Data type: UTCTimestamp</p>

Table 5. Message header tags in requests from a FIX client to TT FIX Adapter

Message header tags in responses from TT FIX Adapter to a FIX client

Tag #	Field name	Req'd	Comments
8	BeginString	Y	<p>FIX protocol version. The tag indicates the beginning of a new message. This tag is always the first tag in the message.</p> <p>The value is always FIX.4.2.</p> <p>Data type: String</p>

Table 6. Message header tags in responses from TT FIX Adapter to a FIX client

Tag #	Field name	Req'd	Comments
9	BodyLength	Y	Message length (in characters), which represents number of characters in the message following this tag up to, and including, the delimiter immediately preceding Tag 10 (Checksum). This tag must be the second field in a message. Data type: int
35	MsgType	Y	Specifies the type of message contained in the message body. This tag must appear third in the list of header tags. Data type: String
49	SenderCompID	Y	TT FIX Adapter ID, as specified as LocalCompID in the TT FIX Adapter configuration. Maximum supported length = 255 characters Data type: String For more information, see the <i>TT FIX Adapter System Administration Manual</i> .
56	TargetCompID	Y	FIX client TT Universal Login ID Maximum supported length = 11 characters Data type: String
34	MsgSeqNum	Y	Message sequence number. Data type: SeqNum
50	SenderSubID	Y	One of the following: <ul style="list-style-type: none"> • If the FIX client request included Tag 57 (TargetSubID), returns that value in this tag. • Otherwise, returns the sender MGT in this tag. Data type: String For more information about the interaction between tags 50, 57 and 116, see the section called Requests and responses involving Tag 50 (SenderSubID), Tag 57 (TargetSubID), Tag 116 (OnBehalfOfID), and Tag 129 (DeliverToSubID) , on page 34.
57	TargetSubID	C	Either the value of Tag 50 (SenderSubID) in the FIX client request, or the TraderIdType value provided by the TT Synthetic Strategy Engine. Data type: String Condition: Sent only if the FIX client request included Tag 50 (SenderSubID), or if this message is for a child order or fill of a synthetic order from a TT Synthetic Strategy Engine with the TraderIdType parameter set. For more information about the interaction between tags 50, 57 and 116, see the section called Requests and responses involving Tag 50 (SenderSubID), Tag 57 (TargetSubID), Tag 116 (OnBehalfOfID), and Tag 129 (DeliverToSubID) , on page 34.

Table 6. Message header tags in responses from TT FIX Adapter to a FIX client

Tag #	Field name	Req'd	Comments
129	DeliverToSubID	C	<p>Value of Tag 116 (OnBehalfOfSubID) in the FIX client request.</p> <p>Data type: String</p> <p>Condition: Sent with the corresponding request contained Tag 116 (OnBehalfOfSubID)</p> <p>Additional Information:</p> <p>For more information about the interaction between tags 50, 57 and 116, see the section called Requests and responses involving Tag 50 (SenderSubID), Tag 57 (TargetSubId), Tag 116 (OnBehalfOfID), and Tag 129 (DeliverToSubID), on page 34.</p>
43	PossDupFlag	C	<p>Whether the sequence number for this message is already used. Possible values include:</p> <ul style="list-style-type: none"> Y: Possible duplicate N: Original transmission <p>Data type: Boolean</p> <p>Condition: Sent when TT FIX Adapter resends messages</p>
97	PossResendFlag	C	<p>Whether the TT FIX Adapter might have already sent this message. When TT FIX Adapter encounters a corrupt FIX message cache during startup, it moves the FIX message cache to a new location and shuts down. When it restarts without the FIX message cache, TT FIX Adapter has no record of the fills it previously sent, so resends them with 97=Y to indicate that the fills might have been sent previously.</p> <p>Possible values include:</p> <ul style="list-style-type: none"> Y: Possible duplicate N: Original transmission <p>Data type: Boolean</p> <p>Condition: Sent when TT FIX Adapter restarts after encountering a corrupt FIX message cache, and only until it completes the initial download.</p>
122	OrigSendingTime	C	<p>Original time of message transmission, when transmitting orders as the result of a resend request. Always expressed in UTC.</p> <p>Data type: UTCTimestamp</p> <p>Condition: Sent when TT FIX Adapter resends a message</p> <p>Note: Some Exchanges might send milliseconds in certain circumstances, in the form: <i>YYYYMMDD-HH:MM:SS.sss</i>. TT recommends that you allocate enough memory to handle the longer timestamp should an Exchange send it.</p>
52	SendingTime	Y	<p>Time, in UTC, the message was sent.</p> <p>Data type: UTCTimestamp</p> <p>Note: Some Exchanges might send milliseconds in certain circumstances, in the form: <i>YYYYMMDD-HH:MM:SS.sss</i>. TT recommends that you allocate enough memory to handle the longer timestamp should an Exchange send it.</p>

Table 6. Message header tags in responses from TT FIX Adapter to a FIX client

Requests and responses involving Tag 50 (SenderSubID), Tag 57 (TargetSubId), Tag 116 (OnBehalfOfID), and Tag 129 (DeliverToSubID)

The following table shows how Tags 50, 57, 116, and 129 interact with each other in the order-related messages sent by a FIX client and the acknowledgements sent by TT FIX Adapter.

Tag 35=	FIX client sends to TT FIX Adapter	TT FIX Adapter forwards to TT Gateway	TT FIX Adapter sends to FIX client
<i>When the FIX client sends Tag 50 without Tag 116</i>			
D, F, or G	50=abc; 116 omitted	Unique Trader ID=abc	Execution Report (8) with 57=abc,129 omitted
<i>When the FIX client sends Tag 50 and Tag 116</i>			
D, F, or G	50=abc; 116=xyz	Unique Trader ID=xyz	For the message sender, Execution Report (8) with 57=abc,129=xyz For all others, Execution Report (8) with 57=xyz
<i>When the FIX client omits Tag 50 and sends Tag 116</i>			
D, F, or G	50 omitted; 116=xyz	Unique Trader ID=xyz	For the message sender, Execution Report (8) with 57=NONE,129=xyz For all others, Execution Report (8) with 57=xyz
<i>When the FIX client omits Tag 50 and Tag 116</i>			
D	50 omitted; 116 omitted	N/A	Execution Report (8) with 57=NONE
F or G	50 omitted; 116 omitted	Unique Trader ID = same value as that from most recent D or G message	For the message sender, Execution Report (8) with tags 57 and 129 equal to the values of tags 50 and 116, respectively, from the most recent D or G message For all others, Execution Report (8) with tag 57 equal to the value of tag 116, if not NONE, or tag 50 otherwise, from the most recent D or G message
<i>When the FIX client sends Tag 57</i>			
D, F, or G	57=xyz		Execution Report (8) with 50=xyz
UAN (for 16724=0,1,4,5,6)	57=xyz		Position Report (UAP) with 50=xyz
<i>When the FIX client omits Tag 57</i>			
D, F, or G	57 omitted		Execution Report (8) with 50=MGT (Member/Group/Trader)
UAN (for 16724=0,1,4,5,6)	57=omitted		Position Report (UAP) with 50=MGT (Member/Group/Trader)

Table 7. Requests involving tags 50, 57, 116, and 129

Note: When routing orders on behalf of multiple traders through a FIX session, the regulations for some exchanges (such as CME and CBOT) require that a unique trader identifier be provided for each trader. Tag 50 (**senderSubID**) or Tag 116 (**onBehalfOfSubID**) provide a means of complying with these regulations.

Message structure: message body

About the message body	The message body contains data specific to the particular message.
Common components in messages	<p>Many messages share a common set of tags. For example, all of the order-related messages contain tags that identify the trader and trading account information. To help identify these shared components, TT FIX Adapter groups the shared tags into the following groups:</p> <ul style="list-style-type: none">• Instrument component block, which contains FIX tags that identify tradeable instruments• Underlying instrument component block, which contains FIX tags that identify legs of multi-leg instruments• Trader component block, which contains FIX tags that identify a trader and corresponding account information
Component differences between request and response	<p>Component blocks can also differ based on the source of the message, whether a request from a FIX client or a response from TT FIX Adapter. For example, an instrument component block comprises FIX tags that fully define a product, including its identity, characteristics, and price data. When submitting a request, you need to include only enough tags in the instrument component block to identify uniquely the instrument. When TT FIX Adapter sends the response, it includes all of the tags - unlike the request which includes only enough tags to uniquely identify the instrument with the requested information.</p>
Component blocks	<p>Each FIX message can contain zero or more component blocks. The FIX Protocol uses the component blocks as a convenient way to group a set of FIX tags that might appear in multiple messages.</p> <p>TT FIX Adapter supports the following component blocks in a message body:</p> <ul style="list-style-type: none">• Instrument component block for requests• Instrument component block for responses• Underlying instrument component block for requests• Underlying instrument component block for responses• Trader component block

Component block: Instrument (FIX client request)

Overview

When sending the instrument component block in a FIX client request, you can specify the instrument in any of the following methods.

- By security ID
- By security Alt ID
- By security name

These methods can be used as follows:

- You can use the security ID method for all cases.
- You can only use the security Alt ID method for contracts for which the exchange populates Tag 10455 (**SecurityAltID**).
- You can use the security name method for all contracts on all exchanges with the following exceptions:
 - You cannot use the security name method for MEFF; you must use the security ID method.
 - You cannot use the security name method for ICE contracts that use daily, weekly, variable, and undefined delivery terms.
 - You cannot use the security name method for BVMF's rollover contracts, which are multi-legged instruments that the exchange lists as futures contracts.

A special note about Tag 55 (Symbol)

The value TT FIX Adapter uses in Tag 55 (**Symbol**) is provided directly from the exchange on which the contract is traded. In the following special cases, TT Gateways concatenate native exchange data to the standard symbol to provide a unique identity for each contract:

- For all products traded on the TT NYSE_Liffe Gateway with the following exchange codes, the TT Gateway prepends the lowercase version of this code to their exchange product symbols.
 - Derivative Products on Euronext Brussels Equities - "B"
 - Derivative Products on Euronext Brussels Indices - "F"
 - Derivative Products on Euronext Paris Fixed Income - "E"
 - Derivative Products on Euronext Paris Indices - "J"
 - Derivative Products on Euronext Paris Equities - "P"
 - Derivative Products on Euronext Paris Commodities - "Y"
 - Derivative Products on Euronext Lisbon Equities - "S"
 - Derivative Products on Euronext Lisbon Indices - "M"
 - Derivative Products on Euronext Amsterdam Equities - "A"
 - Derivative Products on Euronext Amsterdam Indices - "K"
 - Derivative Products on Euronext Amsterdam Currency - "Z"
 - Derivative Products on Euronext Amsterdam Agricultural - "R"
 - Cash Underlying Products on Euronext Amsterdam - "G"
 - Cash Underlying Products on Euronext Paris - "C"
 - Cash Underlying Products on Euronext Lisbon - "H"
 - Cash Underlying Products on Euronext Brussels - "D"
- For all products traded on the SFE trading venue, the TT Gateway prepends

either **CFD**, **SFE**, or **NZFOE** to their exchange product symbols. In addition, the TT SFE Gateway appends an **O** to its exchange product symbols for all overnight option products.

When specifying by security ID Instead of including the entire instrument block when communicating with TT FIX Adapter, client applications can use Tags 48 (**SecurityID**), 55 (**Symbol**), and 207 (**SecurityExchange**). As instruments on different exchanges can share the same value for Security ID, including Security Exchange creates a unique identifier for each instrument.

Tag #	Field name	Req'd	Comments
207	SecurityExchange	Y	Name of the market where the instrument trades. TT FIX Adapter uses this value to identify a security. Data type: Exchange For a list of supported markets, refer to the section called Tag 207 (SecurityExchange) , on page 206.
48	SecurityID	Y	TT security ID Data type: String
55	Symbol	Y	Exchange-provided product symbol for the tradable product. Data type: String.

Table 8. Message body tags in requests from a FIX client to TT FIX Adapter

When specifying by alternate security ID For those exchanges that support alternate security ID (sometimes called aliases), client applications can use Tag 10455 (**SecurityAltID**) to specify its alternate security ID. As instruments on different exchanges can share the same value for SecurityAltID, including SecurityExchange creates a unique identifier for each instrument.

Tag #	Field name	Req'd	Comments
207	SecurityExchange	Y	Name of the market where the instrument trades. TT FIX Adapter uses this value to identify a security. Data type: Exchange For a list of supported markets, refer to the section called Tag 207 (SecurityExchange) , on page 206.
10455	SecurityAltID	Y	Alternate ID for an instrument or security, typically for display purposes. Data type: String
55	Symbol	Y	Exchange-provided product symbol for the tradable product. Data type: String.

Table 9. Message body tags in requests from a FIX client to TT FIX Adapter

Tag #	Field name	Req'd	Comments
167	SecurityType	Y	Asset class of the instrument. Possible values include: CS : common stock FOR : foreign exchange FXNDF : foreign exchange non-deliverable forward FUT : future GOVT : sovereign debt IDX : equity index (valid for market data purposes only) MLEG : multi-leg NRG : energy OPT : option Data type: String

Table 9. Message body tags in requests from a FIX client to TT FIX Adapter

When specifying by security name

You can also identify instruments by specifying its name and characteristics. You might use this method if your application or business practices do not use security IDs.

Tag #	Field name	Req'd	Comments
207	SecurityExchange	Y	Name of the market where the instrument trades. TT FIX Adapter uses this value to identify a security. Data type: Exchange For a list of supported markets, refer to the section called Tag 207 (SecurityExchange) , on page 206.
167	SecurityType	Y	Asset class of the instrument. Possible values include: CS : common stock FOR : foreign exchange FXNDF : foreign exchange non-deliverable forward FUT : future GOVT : sovereign debt IDX : equity index (valid for market data purposes only) MLEG : multi-leg NRG : energy OPT : option Data type: String
55	Symbol	C	Exchange-provided product symbol for the tradable product. Data type: String. Condition: Not used when Tag 167 (SecurityType)= MLEG .
200	MaturityMonthYear	C	Month and year the instrument reaches maturity in the format YYYYMM . Data type: month-year Condition: Required when Tag 167 (SecurityType) is FUT , NRG , or OPT
205	MaturityDay	C	Day of expiration for the instrument. TT FIX Adapter uses this value and Tag 200 (MaturityMonthYear) to specify the maturity date when Tag 167 (SecurityType) is FUT , OPT , or NRG . Data type: day-of-month; Range: 1-31 Condition: Required when multiple contracts exist for the same month

Table 10. Message body tags in requests from a FIX client to TT FIX Adapter

Tag #	Field name	Req'd	Comments
63	SettlType	C	<p>Expiration time for FXNDF contract. Possible values include:</p> <p>Dx: FX tenor expression for "days," where x is any integer greater than 0</p> <p>W: FX tenor expression for "weeks," where x is any integer greater than 0</p> <p>Mx: FX tenor expression for "months," where x is any integer greater than 0</p> <p>Yx: FX tenor expression for "years," where x is any integer greater than 0</p> <p>Data type: String</p> <p>Condition: Required when Tag 167 (SecurityType)=FXNDF</p> <p>Note: Like futures, FXNDF contracts have an expiration. While a futures contract uses Tag 200 (MaturityMonthYear) to indicate the expiration date, FXNDF contracts use this tag to express the time until expiration. Consequently, an FXNDF contract requires the combination of Tag 55 (Symbol), Tag 167 (SecurityType), and Tag 63 (SettlType) to uniquely identify it</p>
18211	ContractTerm	C	<p>Term of delivery for the instrument. TT FIX Adapter uses this value to identify contracts that do not have a monthly delivery term. Possible values include:</p> <p>Q: Quarterly term</p> <p>S: Seasonal term</p> <p>Y: Yearly calendar term</p> <p>Data type: char</p> <p>Conditions: Required when both of the following are true:</p> <ul style="list-style-type: none"> • Tag 167 (SecurityType) is NRG. • The delivery term is not monthly.
201	PutOrCall	C	<p>Whether the option represents a put or call. Possible values include:</p> <p>0: Put</p> <p>1: Call</p> <p>Data type: int</p> <p>Condition: Required when Tag 167 (SecurityType) is OPT</p>
202	StrikePrice	C	<p>Strike price for an option</p> <p>Data type: Price</p> <p>Condition: Required when Tag 167 (SecurityType) is OPT</p>
206	OptAttribute	C	<p>Additional information about the option contract.</p> <p>Data type: char</p> <p>Conditions: Required when both of the following are true:</p> <ul style="list-style-type: none"> • Tag 167 (SecurityType) is OPT. • A version for the underlying instrument exists.

Table 10. Message body tags in requests from a FIX client to TT FIX Adapter

Component block: Instrument (TT FIX Adapter response)

Overview When TT FIX Adapter returns an instrument component block in a response, it includes tags that provide additional information about the instrument.

Supported tags

Tag #	Field name	Req'd	Comments
207	SecurityExchange	Y	Name of the market where the instrument trades. TT FIX Adapter uses this value to identify a security. Data type: Exchange For a list of supported markets, refer to the section called Tag 207 (SecurityExchange) , on page 206.
167	SecurityType	Y	Asset class of the instrument. Possible values include: CS : common stock FOR : foreign exchange FXNDF : foreign exchange non-deliverable forward FUT : future GOVT : sovereign debt IDX : equity index (valid for market data purposes only) MLEG : multi-leg NRG : energy OPT : option Data type: String
55	Symbol	Y	Exchange-provided product symbol for the tradable product. Data type: String
200	MaturityMonthYear	C	Month and year the instrument reaches maturity in the format <i>YYYYMM</i> . Data type: month-year Condition: Sent when Tag 167 (SecurityType) is FUT , OPT , or NRG
205	MaturityDay	C	Day of expiration for the instrument. TT FIX Adapter uses this value and Tag 200 (MaturityMonthYear) to specify the maturity date when Tag 167 (SecurityType) is FUT , OPT , or NRG . Data type: day-of-month; Range: 1-31 Condition: Sent when available

Table 11. Message body tags in responses from TT FIX Adapter to a FIX client

Tag #	Field name	Req'd	Comments
63	SettlType	C	<p>Expiration time for FXNDF contract. Possible values include:</p> <p>Dx: FX tenor expression for "days," where x is any integer greater than 0</p> <p>Wx: FX tenor expression for "weeks," where x is any integer greater than 0</p> <p>Mx: FX tenor expression for "months," where x is any integer greater than 0</p> <p>Yx: FX tenor expression for "years," where x is any integer greater than 0</p> <p>Data type: String</p> <p>Condition: Sent when Tag 167 (SecurityType)=FXNDF</p> <p>Note: Like futures, FXNDF contracts have an expiration. While a futures contract uses Tag 200 (MaturityMonthYear) to indicate the expiration date, FXNDF contracts use this tag to express the time until expiration. Consequently, an FXNDF contract requires the combination of Tag 55 (Symbol), Tag 167 (SecurityType), and Tag 63 (SettlType) to uniquely identify it</p>
18211	ContractTerm	C	<p>Term of delivery for the instrument. TT FIX Adapter uses this value to identify contracts that do not have a monthly delivery term. Possible values include:</p> <p>A: Same day</p> <p>B: Balance of month</p> <p>D: Daily term</p> <p>L: Balance of week</p> <p>N: Next day</p> <p>Q: Quarterly term</p> <p>S: Seasonal term</p> <p>V: Variable term</p> <p>W: Weekly term</p> <p>X: Custom</p> <p>Y: Yearly calendar term</p> <p>Data type: char</p> <p>Conditions: Required when both of the following are true:</p> <ul style="list-style-type: none"> • Tag 167 (SecurityType) is NRG. • The delivery term is not monthly.
201	PutOrCall	C	<p>Whether the option represents a put or call. Possible values include:</p> <p>0: Put</p> <p>1: Call</p> <p>Data type: int</p> <p>Condition: Sent when Tag 167 (SecurityType) is OPT</p>
202	StrikePrice	C	<p>Strike price for an option</p> <p>Data type: Price</p> <p>Condition: Sent when Tag 167 (SecurityType) is OPT</p>
206	OptAttribute	C	<p>Additional information about the option contract.</p> <p>Data type: char</p> <p>Conditions: Required when both of the following are true:</p> <ul style="list-style-type: none"> • Tag 167 (SecurityType) is OPT. • A version for the underlying instrument exists.

Table 11. Message body tags in responses from TT FIX Adapter to a FIX client

Tag #	Field name	Req'd	Comments
15	Currency	Y	ISO-standard symbol for the instrument's trading currency. Data type: Currency Note: Check TT Guardian for a list of currencies supported in the trading environment.
48	SecurityID	Y	TT security ID Data type: String
10455	SecurityAltID	C	Alternate ID for an instrument or security, typically for display purposes. Data type: String Condition: Sent when provided by the TT Gateway Note: If you develop FIX clients that use the TT ICE Gateway, refer to the ICE Notes section, on page 234.

Table 11. Message body tags in responses from TT FIX Adapter to a FIX client

Component block: Underlying instrument

Overview

The underlying instrument component block uses the same structure as the instrument component block (with the TT FIX Adapter source), except that all of its tags represent the underlying variations. For example, instead of Tag 55 (**Symbol**) this block contains Tag 311 (**UnderlyingSymbol**).

This block depends on its parent instrument component block. Whether you need to specify an underlying instrument component block depends on the following:

- If you specified the instrument block by security ID or security AltID, you do not specify the underlying instrument block.
- If you specified the instrument block by name, then you must specify the underlying instrument.

Note: You cannot specify underlying instrument blocks in **Market Data Request (v)** messages

When specifying a security by name

Tag #	Field name	Req'd	Comments
10762	SecuritySubType	Y	Strategy type Data type: String For a list of possible values, refer to the section called Enumerations , on page 206.
146	NoRelatedSym	Y	Number of underlying instruments contained in this repeating group. Data type: NumInGroup
Underlying instrument repeating group			
308	UnderlyingSecurityExchange	Y	Name of the exchange (or market) where the underlying instrument trades. Data type: Exchange For a list of supported markets, refer to Appendix A, Enumerations and Codes .
310	UnderlyingSecurityType	Y	Asset class of the underlying instrument. Possible values include: CS : common stock FXNDF : foreign exchange non-deliverable forward FOR : foreign exchange FUT : future GOVT : sovereign debt OPT : option NRG : energy Data type: String
311	UnderlyingSymbol	Y	Exchange-provided product symbol for the underlying product. Data type: String Note: This tag must appear first in each repeating group.

Table 12. Underlying instrument component block in FIX client requests

Tag #	Field name	Req'd	Comments
313	<code>UnderlyingMaturityMonthYear</code>	C	Month and year the underlying instrument reaches maturity in the format <code>YYYYMM</code> . Data type: month-year Condition: Required when Tag 310 (<code>UnderlyingSecurityType</code>) is FUT , OPT , or NRG
314	<code>UnderlyingMaturityDay</code>	C	Day of expiration for the underlying instrument. FIX Adapter uses this value and Tag 200 (<code>MaturityMonthYear</code>) to specify the maturity date when Tag 310 (<code>UnderlyingSecurityType</code>) is FUT , OPT , or NRG . Data type: day-of-month; Range: 1-31 Condition: Required for TT LME Gateway instruments; ignored for others
18212	<code>UnderlyingContractTerm</code>	C	Term of delivery for the underlying instrument. TT FIX Adapter uses this value to identify contracts that do not have a monthly delivery term. Possible values include: Q : Quarterly term S : Seasonal term Y : Yearly calendar term Data type: char Conditions: Required when both of the following are true: <ul style="list-style-type: none"> Tag 167 (<code>SecurityType</code>) is MLEG. The delivery term is not monthly. Note: Do not send this tag for underlying contracts with a monthly delivery term.
315	<code>UnderlyingPutOrCall</code>	C	Whether the underlying option represents a put or call. Possible values include: 0 : Put 1 : Call Data type: int Condition: Required when Tag 310 (<code>UnderlyingSecurityType</code>) is OPT
316	<code>UnderlyingStrikePrice</code>	C	Strike price for the underlying option Data type: Price Condition: Required when Tag 310 (<code>UnderlyingSecurityType</code>) is OPT
317	<code>UnderlyingOptAttribute</code>	C	Additional information about the underlying options contract. Data type: char Condition: Required when a version for the underlying instrument exists
16624	<code>LegSide</code>	Y	Side of this individual leg within the multi-legged instrument's definition, as defined by the exchange. Possible values include: 1 : Buy 2 : Sell Data type: int
10566	<code>LegPrice</code>	C	Price of a futures leg in the strategy. Data type: Price Condition: Required for futures leg of covered strategies, such as volatility trades.

Table 12. Underlying instrument component block in FIX client requests

Tag #	Field name	Req'd	Comments
319	RatioQty	Y	<p>One of the following:</p> <ul style="list-style-type: none"> For a leg of a covered strategy (such as a volatility trade) on CME, CBOT, eCBOT, or NYSE_Liffe markets, the value represents the delta (expressed as an integer between 1 and 100). In all other cases, the value represents the quantity of this leg in the strategy. <p>Data type: Qty</p>

Table 12. Underlying instrument component block in FIX client requests

**When receiving the block
in TT FIX Adapter
response**

Tag #	Field name	Req'd	Comments
10762	SecuritySubType	Y	<p>Strategy type</p> <p>Data type: String</p> <p>For a list of possible values, refer to the section called Enumerations, on page 206.</p>
146	NoRelatedSym	Y	<p>Number of underlying products contained in this repeating group.</p> <p>Data type: NumInGroup</p>
Underlying instrument repeating group			
308	UnderlyingSecurityExchange	Y	<p>Name of the exchange (or market) where the underlying instrument trades.</p> <p>Data type: Exchange</p> <p>For a list of supported markets, refer to Appendix B, Supported FIX Messages and Tags, on page 211.</p>
310	UnderlyingSecurityType	Y	<p>Asset class of the underlying instrument. Possible values include:</p> <ul style="list-style-type: none"> CS: common stock FOR: foreign exchange FUT: future OPT: option NRG: energy <p>Data type: String</p>
311	UnderlyingSymbol	Y	<p>Exchange-provided product symbol for the underlying product.</p> <p>Data type: String</p> <p>Note: This tag must appear first in each repeating group.</p>
313	UnderlyingMaturityMonthYear	C	<p>Month and year the underlying instrument reaches maturity in the format YYYYMM.</p> <p>Data type: month-year</p> <p>Condition: Sent when Tag 310 (UnderlyingSecurityType) is FUT, OPT, or NRG</p>

Table 13. Underlying instrument component block in TT FIX Adapter responses

Tag #	Field name	Req'd	Comments
314	<code>UnderlyingMaturityDay</code>	C	Day of expiration for the underlying instrument. FIX Adapter uses this value and Tag 200 (<code>MaturityMonthYear</code>) to specify the maturity date when Tag 310 (<code>UnderlyingSecurityType</code>) is FUT , OPT , or NRG . Data type: day-of-month; Range: 1-31 Condition: Sent when available from the exchange
18212	<code>UnderlyingContractTerm</code>	C	Term of delivery for the underlying instrument. TT FIX Adapter uses this value to identify contracts that do not have a monthly delivery term. Possible values include: D : Daily term W : Weekly term Q : Quarterly term S : Seasonal term Y : Yearly calendar term V : Variable term Data type: char Conditions: Sent when both of the following are true: <ul style="list-style-type: none"> Tag 167 (<code>SecurityType</code>) is MLEG. The delivery term is not monthly.
315	<code>UnderlyingPutOrCall</code>	C	Whether the underlying option represents a put or call. Possible values include: 0 : Put 1 : Call Data type: int Condition: Sent when Tag 310 (<code>UnderlyingSecurityType</code>) is OPT
316	<code>UnderlyingStrikePrice</code>	C	Strike price for the underlying option Data type: Price Condition: Sent when Tag 310 (<code>UnderlyingSecurityType</code>) is OPT
317	<code>UnderlyingOptAttribute</code>	C	Additional information about the underlying options contract. Data type: char Condition: Required when a version for the underlying instrument exists
318	<code>UnderlyingCurrency</code>	C	ISO-standard symbol for the underlying instrument's trading currency. Data type: Currency Note : Check TT Guardian for a list of currencies supported in the trading environment. Condition: Included in Execution Report (8) and Position Report (uAP) messages when the TT FIX Adapter configuration enables the Send Security Legs settings for the FIX Session
309	<code>UnderlyingSecurityID</code>	Y	Exchange-provided identifier for the underlying tradable product. Data type: String Condition: Sent when provided by the TT Gateway.
10456	<code>UnderlyingSecurityAltID</code>	C	Alternate ID for the underlying instrument or security, typically for display purposes. Data type: String

Table 13. Underlying instrument component block in TT FIX Adapter responses

Tag #	Field name	Req'd	Comments
54	Side	C	Side of the order. Possible values include: 1: Buy 2: Sell Data type: char Condition: Sent for all Security Definition (d) messages. For all other messages, TT FIX Adapter sends the information in Tag 16624 (LegSide).
16624	LegSide	Y	Side of this individual leg within the multi-legged instrument's definition, as defined by the exchange. Possible values include: 1: Buy 2: Sell Data type: int Condition: Sent for all messages other than Security Definition (d) . For Security Definition (d) messages, TT FIX Adapter sends the information in Tag 54 (Side).
10566	LegPrice	C	Price of a futures leg in the strategy. Data type: Price Condition: Sent for futures leg of covered strategies, such as volatility trades.
319	RatioQty	Y	One of the following: <ul style="list-style-type: none"> For a leg of a covered strategy (such as a volatility trade) on CME, CBOT, eCBOT, or NYSE_Liffe markets, the value represents the delta (expressed as an integer between 1 and 100). In all other cases, the value represents the quantity of this leg in the strategy. Data type: Qty

Table 13. Underlying instrument component block in TT FIX Adapter responses

Component block: Trader

Overview Certain message types include a trader component block, which carries trader and account information.

Tags in messages sent from FIX clients to TT FIX Adapter The Trader Component Block uses the **Account Defaults** settings defined in TT User Setup for a user to determine which FIX tags are required for a particular account/market/product type.

For more information about configuring Account Defaults, refer to the TT User Setup help.

Tag #	Field name	Req'd	Comments
1	Account	Y	Order-routing account Data type: String
440	ClearingAccount	C	Clearing account Data type: String Conditions: Required if both of the following are true: <ul style="list-style-type: none"> The TT Gateway receiving the order requires a clearing account (Give Up Member) ID. TT User Setup does not specify a default Give Up Member value in the Account Defaults settings for the user. For more information, see the TT User Setup online help.
16102	FFT2	C	TT FFT2 field Data type: String Conditions: Required if both of the following are true: <ul style="list-style-type: none"> The TT Gateway receiving the order requires an FFT2 value. TT User Setup does not specify a default FFT2 value in the Account Defaults settings for the user. For more information, see the TT User Setup online help.
16103	FFT3	C	TT FFT3 field Data type: String Conditions: Required if both of the following are true: <ul style="list-style-type: none"> The TT Gateway receiving the order requires an FFT3 value. TT User Setup does not specify a default FFT3 value in the Account Defaults settings for the user. For more information, see the TT User Setup online help.

Table 14. Trader component block tags in FIX client requests

Tag #	Field name	Req'd	Comments
47 204 or 18205	Rule80A CustomerOrFirm or TTAccountType	C	<p>TT origin code</p> <p>Data type: String</p> <p>Condition: Required if both of the following are true:</p> <ul style="list-style-type: none"> The TT Gateway receiving the order requires an account type. TT User Setup does not specify a default Account Type value in the Account Defaults settings for the user. <p>Additional Information:</p> <p>When required, the message must include either the combination of Tags 47 (Rule80A) and 204 (CustomerOrFirm) or Tag 18205 (TTAccountType), as follows:</p> <ul style="list-style-type: none"> First Agent Account (A1): Set 47=A and 204=0, or set 18205=A1 Second Agent Account (A2): Set 47=A and 204=1, or set 18205=A2 Third Agent Account (A3): Set 47=A and 204=2, or set 18205=A3 Fourth Agent Account (A4): Set 47=A and 204=3, or set 18205=A4. Fifth Agent Account (A5): Set 47=A and 204=4, or set 18205=A5. Sixth Agent Account (A6): Set 47=A and 204=5, or set 18205=A6. Seventh Agent Account (A7): Set 47=A and 204=6, or set 18205=A7. Eighth Agent Account (A8): Set 47=A and 204=7, or set 18205=A8. Ninth Agent Account (A9): Set 47=A and 204=8, or set 18205=A9. Pre-Designated Give-up Trade (G1): Set 47=W and 204=0, or set 18205=G1 Designated Give-up Trade/Automatic Allocation (G2): Set 47=W and 204=1, or set 18205=G2 Give-up System Allocation (G3): Set 47=W and 204=2, or set 18205=G3 First Market Maker Account (M1): Set 47=E and 204=0, or set 18205=M1 Second Market Maker Account (M2): Set 47=E and 204=1, or set 18205=M2 Market Maker/Give-up or System Allocation (M3): Set 47=E and 204=2, or set 18205=M3 First Principal Account (P1): Set 47=P and 204=0, or set 18205=P1 Second Principal Account (P2): Set 47=P and 204=1, or set 18205=P2 House/Give-up or System Allocation (P3): Set 47=P and 204=2, or set 18205=P3 Unallocated (U1): Set 47=0 and 204=0, or set 18205=U1 (For orders that have not been allocated to a customer account or where allocation is a middle/back office function) Unallocated/Automatic (U2): Set 47=0 and 204=1, or set 18205=U2 Unallocated/System (U3): Set 47=0 and 204=2, or set 18205=U3 <p>For more information, see the TT User Setup online help.</p> <p>Note: TT FIX Adapter7.4.1 marks the beginning of the deprecation process for Tag 47 (Rule80A) and Tag 204 (CustomerOrFirm). You should transition to using Tag 18205 (TTAccountType).</p>

Tags in messages sent from TT FIX Adapter to FIX clients

The TT FIX Adapter uses the global FIX session settings in the configuration to determine which TT fields to include in the FIX tags of the Trader Component Block in message sent from TT FIX Adapter to FIX clients.

Tag #	Field name	Req'd	Comments
1	Account	C	Order-routing account. The value matches the Tag 1 (Account) in the corresponding request. Data type: String Conditions: <ul style="list-style-type: none"> • Sent in Execution Report (8) messages, except for those in response to an Order Status Request (8) where no matching orders can be found. • Sent in Position Report (UAP) messages only when Tag 16727 (TotalNumPosReports) > 0 and Tag 16724 (PosReqType) = 0 (positions), 1 (trades), 5 (manual fills), or 6 (detailed SODs).
440	ClearingAccount	C	Clearing account Data type: String Condition: Sent when the tag contains a value.
16102	FFT2	C	TT FFT2 field Data type: String Condition: Sent when the tag contains a value.
16103	FFT3	C	TT FFT3 field Data type: String Condition: Sent when the tag contains a value.

Table 15. Trader component block tags in TT FIX Adapter responses

Tag #	Field name	Req'd	Comments
47 204 or 18205	Rule80A CustomerOrFirm or TTAccountType	C	<p>TT origin code</p> <p>Data type: String</p> <p>Condition: Based on the value of the Send Account Type As option in the Global FIX session settings in the configuration:</p> <ul style="list-style-type: none"> • Tags 47 and 204. Sends Tags 47 (Rule80A) and 204 (CustomerOrFirm), but not Tag 18205 (TTAccountType). • Tag 18205. Sends Tag 18205 (TTAccountType), not Tag 47 (Rule80A) or Tag 204 (CustomerOrFirm). • Don't Send. Sends none of these tags. <p>Additional Information:</p> <p>When sent, the message must include either the combination of Tags 47 (Rule80A) and 204 (CustomerOrFirm) or Tag 18205 (TTAccountType), as follows:</p> <ul style="list-style-type: none"> • First Agent Account (A1): Set 47=A and 204=0, or set 18205=A1 • Second Agent Account (A2): Set 47=A and 204=1, or set 18205=A2 • Third Agent Account (A3): Set 47=A and 204=2, or set 18205=A3 • Fourth Agent Account (A4): Set 47=A and 204=3, or set 18205=A4 • Fifth Agent Account (A5): Set 47=A and 204=4, or set 18205=A5 • Sixth Agent Account (A6): Set 47=A and 204=5, or set 18205=A6 • Seventh Agent Account (A7): Set 47=A and 204=6, or set 18205=A7 • Eighth Agent Account (A8): Set 47=A and 204=7, or set 18205=A8 • Ninth Agent Account (A9): Set 47=A and 204=8, or set 18205=A9 • Pre-Designated Give-up Trade (G1): Set 47=W and 204=0, or set 18205=G1 • Designated Give-up Trade/Automatic Allocation (G2): Set 47=W and 204=1, or set 18205=G2 • Give-up System Allocation (G3): Set 47=W and 204=2, or set 18205=G3 • First Market Maker Account (M1): Set 47=E and 204=0, or set 18205=M1 • Second Market Maker Account (m2): Set 47=E and 204=1, or set 18205=M2 • Market Maker/Give-up or System Allocation (M3): Set 47=E and 204=2, or set 18205=M3 • First Principal Account (P1): Set 47=P and 204=0, or set 18205=P1 • Second Principal Account (P2): Set 47=P and 204=1, or set 18205=P2 • House/Give-up or System Allocation (P3): Set 47=P and 204=2, or set 18205=P3 • Unallocated (U1): Set 47=0 and 204=0, or set 18205=U1 (For orders that have not been allocated to a customer account or where allocation is a middle/back office function) • Unallocated/Automatic (U1): Set 47=0 and 204=1, or set 18205=U2 • Unallocated/System (U3): Set 47=0 and 204=2, or set 18205=U3 <p>For more information about the Global FIX session options, see the <i>TT FIX Adapter System Administration Manual</i>.</p>

Table 15. Trader component block tags in TT FIX Adapter responses(Continued)

Tag #	Field name	Req'd	Comments
10553	TTUsername	C	TT Universal Login ID Data type: String Condition: Sent only if the user who routed the order was logged in with a TT Universal Login ID.

Table 15. Trader component block tags in TT FIX Adapter responses(Continued)

Understanding Account Defaults

What are Account Defaults?

When TT FIX Adapter receives an order from a FIX client, it maps FIX tags to TT order attributes. By default, the FIX client must provide all data required by the TT Gateway / exchange in the FIX order routing message. You can set up Account Defaults to use default values for certain tags in lieu of including them in every FIX order routing message.

Account Defaults are actually a set of rules. A rule contains two types of information:

- Keys, which TT FIX Adapter uses to match incoming orders to a particular rule
- Default values, which TT FIX Adapter applies to any order that matches a rule and is missing one or more of the values in the FIX order routing message

Specifically, each rule consists of the following fields.

Field	Description	Type	FIX Tag
Account	Customer account number	Key	1 (Account)
Market	TT Market name	Key	*
Gateway	TT Gateway name	Key	*
Product Type	Product type associated with a security	Key	167 (SecurityType)
Account Type	Type of account associated with the trader	Default value	47 (Rule80A) and 204 (CustomerOrFirm) -OR- 18205 (TTAccountType)
Give-Up Member	Member code for the clearing member; used for give-up trades only	Default value	440 (ClearingAccount)
FFT2	Free-form fields, which might be required by some TT Gateways	Default value	16102 (FFT2)
FFT3			16103 (FFT3)

* The way TT FIX Adapter uses the Market and Gateway fields depends on whether the configuration overrides the Market Name for the TT Gateway (for more information about this configuration parameter, see the TT FIX Adapter System Administration Manual), as follows:

- If Tag 207 (**SecurityExchange**) contains a TT Market name, TT FIX Adapter uses the value from the order routing message.
- If Tag 207 (**SecurityExchange**) contains an overridden Market name, TT FIX Adapter uses the Market name corresponding to the destination TT Gateway instead.

Table 16. TT FIX Adapter Account Defaults

The TT User Setup administrator can configure Account Defaults for each user. Refer to the TT User Setup help system for more information.

How TT FIX Adapter uses Account Defaults

When TT FIX Adapter receives an order from a FIX client, it must convert it into a TT order that conforms to the TT Gateway requirements. If the order omits some of this information, TT FIX Adapter tries to match the order with one of the Account Default rules. If it finds a match, TT FIX Adapter inserts any specified default values into the order before routing it to the specified TT Gateway.

Consequently, tags 47 (**Rule80A**), 204 (**CustomerOrFirm**), 18205 (**TTAccountType**), 440 (**ClearingAccount**), 16102 (**FFT2**), and 16103 (**FFT3**) are all optional in a FIX message if a FIX session contains an Account Default with default values.

When TT FIX Adapter receives an order message from a FIX client, it starts the process of matching the order to a set of Account Defaults. It compares the Account Defaults keys to the corresponding tags in the order as follows:

- Account key to Tag 1 (**Account**)
- Market key to Tag 207 (**SecurityExchange**)
- Gateway key to Tag 207 (**SecurityExchange**)

Note: As this tag is optional, TT FIX Adapter internally determines the destination Gateway if the order routing message omits this tag.

- Product Type key to Tag 167 (**SecurityType**)

If the values are the same, TT FIX Adapter considers the order to match the Account Default entry. Note that when you provide multiple Account Defaults with the same keys, TT FIX Adapter matches the first one it finds. If it finds no matching Account Defaults entry, TT FIX Adapter expects the FIX client to provide all of the necessary order information in the FIX message.

Account Defaults matching order

To help you understand the matching process, assume an administrator configures two FIX clients with the same set of Account Defaults, but in a different order. Each FIX client sending the same message to TT FIX Adapter receives different results.

TT User Setup includes the following Account Defaults entries for FIX client ABC:

Account	Market	Gateway	Prod Type	Acct Type	Give-Up	FFT2	FFT3
*	ICE_IPE	*	NRG	G1	999		
*	*	*	*	A1			

For FIX client XYZ, TT User Setup includes the same Account Defaults entries, but in reverse order, as shown:

Account	Market	Gateway	Prod Type	Acct Type	Give-Up	FFT2	FFT3
*	*	*	*	A1			
*	ICE_IPE	*	NRG	G1	999		

Both FIX clients send a message with:

- Tag 1 (**Account**) = "123"
- Tag 207 (**SecurityExchange**) = "ICE_IPE"
- Tag 167 (**SecurityType**) = "NRG"

When TT FIX Adapter receives the message from FIX client ABC, it finds a match in the first Account Defaults listed, so it routes the order to the TT Gateway with Account Type = "G1" and Give-Up = "999". When TT FIX Adapter receives the same message from FIX client XYZ, it matches the first row. Even though the second Account Defaults entry represents a more precise match, TT FIX Adapter stops looking after it finds the first match. Consequently, when TT FIX Adapter receives this message, it routes it to the TT Gateway with Account Type = "A1".

Locking Account Defaults

If you want TT FIX Adapter to limit the ability of a FIX client to submit other values in the TT order fields, you can lock the Account Defaults by enabling the **Restrict user from overriding Account Defaults** setting in TT User Setup.

When you lock the Account Defaults for a FIX session, TT FIX Adapter:

- Rejects any incoming order that does not match an Account Defaults entry.
- Rejects any incoming order that matches an Account Defaults entry but supplies values that differ from the default values specified in the matching entry.

Account Defaults: Order Matching and Routing Examples

About the examples

This section presents the following examples that help illustrate how TT FIX Adapter matches incoming order requests from FIX clients to Account Defaults and how it affects the orders TT FIX Adapter sends to the TT Gateways:

- Example 1 shows a typical order that matches the key values of an Account Defaults entry.
- Example 2 illustrates what happens when an order does not match any Account Defaults entry and does not provide information required by a TT Gateway.
- Example 3 shows an order that matches an Account Defaults entry with wildcards as key values.
- Example 4 uses an order that overrides the default values for a matching Account Defaults entry.
- Example 5 shows what happens when an order tries to override the default values for a matching Account Defaults entry when TT FIX Adapter locks the Account Defaults.
- Example 6 uses an order that complies with the default values for a matching Account Defaults entry when TT FIX Adapter locks the Account Defaults.

Each example shows the contents of the incoming FIX message, the configured Account Defaults, and the resulting TT order. The examples also assume that TT FIX Adapter connects to one flavor of a TT CME Gateway, namely CME-A.

Example 1: order matches an Account Defaults entry

This example shows what happens when a FIX client submits an order request that matches one of the configured Account Defaults. In this example, a TT User Setup administrator configures the following Account Defaults entries for a user and disables **Restrict user from overriding Account Defaults**.

Account	Market	Gateway	Prod Type	Acct Type	Give-Up	FFT2	FFT3
123	CME	*	FUTURE	A3			
123	CME	*	*	A2			
*	*	*	*	A1			

The following table shows an incoming FIX message, which Account Defaults entry TT FIX Adapter selects, and the resulting TT order it sends to the TT Gateway.

FIX Message Tags	Account Defaults	TT Order
1 (Account)	123	123
207 (SecurityExchange)	CME	CME-A
167 (SecurityType)	SPREAD	SPREAD
47 (Rule80A)	Account Type	A2
204 (CustomerOrFirm)		
18205 (TTAccountType)		
440 (ClearingAccount)	Give-Up Member	
16102 (FFT2)	FFT2	
16103 (FFT3)	FFT3	

Example 2: order matches no Account Default entry and omits information required by the TT Gateway

This example shows what happens when a FIX client submits an order request that matches none of the Account Defaults TT User Setup defines for the user. The order also omits the Account Type information, which the TT CME Gateway requires in all order requests. In this example, a TT User Setup administrator configures the following Account Defaults entries for a user and disables **Restrict user from overriding Account Defaults**.

Account	Market	Gateway	Prod Type	Acct Type	Give-Up	FFT2	FFT3
123	CME	*	FUTURE	A3			

The following table shows an incoming FIX message, which Account Defaults TT FIX Adapter selects, and the resulting TT order it sends to the TT Gateway.

FIX Message Tags	Account Defaults	TT Order
1 (Account)	789	789
207 (SecurityExchange)	CME	CME-A
167 (SecurityType)	FUTURE	FUTURE
47 (Rule80A)		
204 (CustomerOrFirm)		
18205 (TTAccountType)		
440 (ClearingAccount)	Give-Up Member	
16102 (FFT2)	FFT2	
16103 (FFT3)	FFT3	

In this example, TT FIX Adapter submits a TT order to the TT CME Gateway with the information included in the FIX message only, because the order did not match any Account Defaults. The CME exchange happens to require an Account Type for all submitted orders. Because the TT order does not contain an Account Type, the TT CME Gateway rejects the order.

Example 3: order matches Account Defaults with wildcard keys

This example shows what happens when a FIX client submits an order request that matches one Account Defaults entry with the wildcard (*) in every key. In this example, a TT User Setup administrator configures the following Account Defaults entries for a user and disables **Restrict user from overriding Account Defaults**.

Account	Market	Gateway	Prod Type	Acct Type	Give-Up	FFT2	FFT3
*	*	*	*	A1			

The following table shows an incoming FIX message, which Account Defaults TT FIX Adapter selects, and the resulting TT order it sends to the TT Gateway.

FIX Message Tags	Account Defaults	TT Order
1 (Account)	123	123
207 (SecurityExchange)	CME	CME-A
167 (SecurityType)	FUTURE	FUTURE
47 (Rule80A)		
204 (CustomerOrFirm)		
18205 (TTAccountType)		

FIX Message Tags	Account Defaults	TT Order
440 (ClearingAccount)	Give-Up Member	
16102 (FFT2)	FFT2	
16103 (FFT3)	FFT3	

Example 4: order overrides default value for unlocked Account Defaults

This example shows what happens when a FIX client submits an order request that matches one of the Account Defaults configured in TT User Setup. It also overrides one of the default values (Acct Type) in the entry. In this example, a TT User Setup administrator configures the following Account Defaults entries for a user and disables **Restrict user from overriding Account Defaults**.

Account	Market	Gateway	Prod Type	Acct Type	Give-Up	FFT2	FFT3
123	CME	*	*	A2			
123	*	*	*	A1			

The following table shows an incoming FIX message, which Account Defaults TT FIX Adapter selects, and the resulting TT order it sends to the TT Gateway.

FIX Message Tags	Account Defaults	TT Order
1 (Account)	123	123
207 (SecurityExchange)	CME	CME-A
167 (SecurityType)	FUTURE	FUTURE
47 (Rule80A)		M1
204 (CustomerOrFirm)		
18205 (TTAccountType)	M1	
440 (ClearingAccount)	Give-Up Member	
16102 (FFT2)	789	789
16103 (FFT3)	FFT3	

Because the FIX session does not lock the Account Defaults, TT FIX Adapter sends the value of Tag 18205 (TTAccountType) in the TT order **Account Type** field.

Example 5: order overrides a default value for locked Account Defaults

This example shows what happens when a FIX client submits an order request that tries to override a default value of a FIX session with locked Account Defaults. This example provides a value in Tag 18205 (TTAccountType) that differs from the Account Type in the matching Account Defaults entry. In this example, a TT User Setup administrator configures the following Account Defaults entries for a user and enables **Restrict user from overriding Account Defaults**.

Account	Market	Gateway	Prod Type	Acct Type	Give-Up	FFT2	FFT3
123	CME	*	FUTURE	A2			
*	*	*	*	A1			

The following table shows an incoming FIX message, which Account Defaults TT FIX Adapter selects, and the resulting TT order it sends to the TT Gateway.

FIX Message Tags		Account Defaults		TT Order
1 (Account)	123	Account	123	123
207 (SecurityExchange)	CME	Market	CME	CME-A
167 (SecurityType)	FUTURE	Product Type	*	FUTURE
47 (Rule80A)		Account Type	A2	ERROR
204 (CustomerOrFirm)				
18205 (TTAccountType)	M1			
440 (ClearingAccount)		Give-Up Member		
16102 (FFT2)		FFT2		
16103 (FFT3)		FFT3		

The FIX session in this example locks the Account Defaults, so FIX clients are not permitted to override. The incoming FIX message supplies "M1" in Tag 18205 (TTAccountType), which differs from the default "A2" value for the Account Type. Consequently, TT FIX Adapter rejects the order directly, without sending it to the TT Gateway.

Example 6: order matches default value for locked Account Defaults

This example shows what happens when a FIX client submits an order request that complies with a default value of a FIX session with locked Account Defaults. This example provides a value in Tag 18205 (TTAccountType) that matches the Account Type in the matching Account Defaults entry. In this example, a TT User Setup administrator configures the following Account Defaults entries for a user and enables **Restrict user from overriding Account Defaults**.

Account	Market	Gateway	Prod Type	Acct Type	Give-Up	FFT2	FFT3
123	CME	*	FUTURE	A2			
*	*	*	*	A1			

The following table shows an incoming FIX message, which Account Defaults TT FIX Adapter selects, and the resulting TT order it sends to the TT Gateway.

FIX Message Tags		Account Defaults		TT Order
1 (Account)	123	Account	123	123
207 (SecurityExchange)	CME	Market	CME	CME-A
167 (SecurityType)	FUTURE	Product Type	FUTURE	FUTURE
47 (Rule80A)		Account Type	A2	A2
204 (CustomerOrFirm)				
18205 (TTAccountType)	A2			
440 (ClearingAccount)		Give-Up Member		
16102 (FFT2)		FFT2		
16103 (FFT3)		FFT3		

The TT FIX Adapter in this example also locks the Account Defaults for the FIX session. However, the value of Tag 18205 (TTAccountType) matches the default value for the Account Type. Therefore, TT FIX Adapter routes the corresponding TT order to the TT Gateway.

Message structure: message footer

Overview Every message sent to and from the TT FIX Adapter must end with a message footer.

Message footer tags The message footer contains the following tags.

Tag #	Field name	Req'd	Comments
10	Checksum	Y	A three-digit character representation used to segregate messages, which is always the last field in a message (i.e. it serves, with the trailing <SOH>, as the end-of-message delimiter). Data type: String

Table 17. Message footer tags

Chapter overview

This chapter describes these session-level messages supported by the TT FIX Adapter.

In this chapter

Section	Page
About session-level messages	64
Logon (A)	65
Logout (5)	67
Heartbeat (0)	68
Test Request (1)	69
Resend Request (2)	70
Sequence Reset (4)	71
Session-Level Reject (3)	72

About session-level messages

Overview Session-level messages communicate the state of the connection between a FIX client and a TT FIX Adapter.

Supported messages Table 18 describes the session-level FIX messages that TT FIX Adapter currently supports.

Message	Purpose
Heartbeat (0)	Used to verify communication with TT FIX Adapter
Test Request (1)	Used to request a heartbeat
Resend Request (2)	Used to request a resend of a range of messages
Session-Level Reject (3)	Used to return the reason for rejecting an incoming message
Sequence Reset (4)	Used to reset the message sequence number or gap fill
Logout (5)	Used to terminate a connection between a FIX client and the TT FIX Adapter
Logon (A)	Used to establish a connection between a FIX client and the TT FIX Adapter

Table 18. Session-level FIX messages

FIX messages and configuration types The following table shows the FIX messages that each type of TT FIX Adapter configuration can exchange with FIX clients.

Use case Session type	Drop Copy		Order Routing	
	Price	Order	Price	Order
Heartbeat (0)	Yes	Yes	Yes	Yes
Test Request (1)	Yes	Yes	Yes	Yes
Resend Request (2)	Yes	Yes	Yes	Yes
Session-Level Reject (3)	Yes	Yes	Yes	Yes
Sequence Reset (4)	Yes	Yes	Yes	Yes
Logout (5)	Yes	Yes	Yes	Yes
Logon (A)	Yes	Yes	Yes	Yes

Table 19. Session-level message mappings

Logon (A)

Purpose Used to establish a connection between a FIX client and the TT FIX Adapter

Message direction From TT FIX Adapter to FIX client and from FIX client to TT FIX Adapter

Message availability

Use Case	Price Session	Order Session
Drop Copy	Yes	Yes
Order Routing	Yes	Yes

Supported tags

Tag #	Field Name	Req'd	Comments
Component Block <Standard Header>		Y	35=A (MsgType)
96	RawData	C	Logon password. The value must match the logon password for the FIX session. The TT FIX Adapter does not include this tag in its responses. Data type: data Condition: Required in FIX client requests.
98	EncryptMethod	Y	Method of encryption. As the TT FIX Adapter does not support encrypted logons, you must set the value to 0 (None/Other). Data type: int
108	HeartBtInt	Y	Heartbeat interval. Data type: int
141	ResetSeqNumFlag	N	Whether to reset the sequence numbers on both sides of the FIX session. Valid values include Y and N (default). Data type: Boolean Note: If the TT FIX Adapter configuration enables the Send Orders On Reset option and you set 141=Y , TT FIX Adapter sends an Execution Report (8) with Tag 150 (ExecType)= D for all working orders immediately after TT FIX Adapter sends the Logon (A) acknowledgement. Enabling this configuration option performs the same function as requesting an order book download using an Order Status Request (H) message without Tag 11 (ClOrdID) and Tag 37 (OrderID). For more information, see the <i>TT FIX Adapter System Administration Manual</i> .
Component Block <Standard Trailer>		Y	

Message notes

The **Logon** (A) message is used by a FIX client to begin a new or continue an existing session with the TT FIX Adapter.

The following data flow illustrates a typical logon sequence between the TT FIX Adapter and FIX client for an order session:

- 1 FIX client initiates a socket connection.
- 2 FIX client sends a **Logon** (A) message to the TT FIX Adapter.
- 3 The TT FIX Adapter receives the FIX client logon message and verifies the following tags in the **Logon** (A) message against the corresponding parameter values:

- Tag 49 (**SenderCompID**) to **RemoteCompId** (in **FACT**)
 - Tag 56 (**TargetCompID**) to **LocalCompId (in TT User Setup)**
 - Tag 96 (**RawData**) to **Password (in TT User Setup)**
- 4 If any of the following errors occur, the TT FIX Adapter stops processing the logon attempt, sends the FIX Client a **Logout (5)** message with the reason for rejection in Tag 58 (**Text**), and closes the socket connection.
- These tags do not match for the adapter and the client.
 - TT User Setup does not respond.
 - The TT universal login credentials (tags 49 and 96) are invalid.
- 5 Examine the value of Tag 141 (**ResetSeqNumFlag**).
- If Tag 141 (**ResetSeqNumFlag**) is set to N, the TT FIX Adapter compares the value of Tag 34 (**MsgSeqNum**) in the **Logon (A)** message against its current sequence number.
- If **MsgSeqNum** is less than the TT FIX Adapter's expected sequence number, the TT FIX Adapter stops processing the logon attempt, sends a **Logout (5)** message as its response, and closes the socket connection.
 - If **MsgSeqNum** is greater than the TT FIX Adapter's expected sequence number, the logon process continues normally. However, after the logon process completes, the TT FIX Adapter sends a **Resend Request (2)** message.
- If Tag 141 (**ResetSeqNumFlag**) is set to Y, the TT FIX Adapter expects the value of Tag 34 (**MsgSeqNum**) in the **Logon (A)** message to be 1.
- If Tag 34 is not 1, TT FIX Adapter stops processing the logon attempt, sends a **Logout (5)** message, and closes the connection.
 - Otherwise, the logon process continues normally.
- 6 TT FIX Adapter sends a **Logon (A)** message to the FIX client.

Related information

[Logout \(5\)](#), on page 67

[Resend Request \(2\)](#), on page 70

[Sequence Reset \(4\)](#), on page 71

FIX Message Conversations, [Logon and logout](#), on page 162

Logout (5)

Purpose Used to terminate a connection between a FIX client and the TT FIX Adapter

Message direction From TT FIX Adapter to FIX client and from FIX client to TT FIX Adapter

Message availability

Use Case	Price Session	Order Session
Drop Copy	Yes	Yes
Order Routing	Yes	Yes

Supported tags

Tag #	Field Name	Req'd	Comments
	Component Block <Standard Header>	Y	35=5 (MsgType)
58	Text	N	Used to store the reason for the logout. Data type: String
18000	ForceLogout	N	Used to indicate whether the logout is forced. Valid values include: 0 : Unforced logout 1 : Forced logout Data type: char
	Component Block <Standard Trailer>	Y	

Message notes The **Logout (5)** message is used by a FIX client or the TT FIX Adapter to terminate their connection.

FIX clients can use Tag 18000 (**ForceLogout**) to determine whether they should try to reconnect to the TT FIX Adapter automatically. If the TT FIX Adapter forced the logout, manual intervention is likely needed before another attempt will succeed.

Related information [Logon \(A\)](#), on page 65

Heartbeat (0)

Purpose Used to verify communication with TT FIX Adapter

Message direction From TT FIX Adapter to FIX client and from FIX client to TT FIX Adapter

Message availability

Use Case	Price Session	Order Session
Drop Copy	Yes	Yes
Order Routing	Yes	Yes

Supported tags

Tag #	Field Name	Req'd	Comments
	Component Block <Standard Header>	Y	35=0 (MsgType)
112	TestReqID	C	Identifier included in the Test Request (1) message. Data type: String Condition: Sent only in a response to a Test Request (1) message, in which case, the value matches Tag 112 (TestReqID) sent in the request.
	Component Block <Standard Trailer>	Y	

Message notes During periods of inactivity, FIX clients should generate a heartbeat message at regular time intervals. The **Heartbeat (0)** message is used by a FIX client to monitor the status of the communication link. You specify the heartbeat interval timer in Tag 108 (**HeartBtInt**) of the **Logon (A)**.

FIX clients should reset the heartbeat interval timer after every transmitted message (not just heartbeats). When either party stops receiving data for the specified heartbeat interval, it should send a **Test Request (1)** message to verify the connection. The counterparty should send a **Heartbeat (0)** message in response. If a party receives no **Heartbeat (0)** message in a reasonable amount of time, the connection should be considered lost.

Related information [Test Request \(1\)](#), on page 69

Test Request (1)

Purpose Used to request a heartbeat

Message direction From TT FIX Adapter to FIX client and from FIX client to TT FIX Adapter

Message availability

Use Case	Price Session	Order Session
Drop Copy	Yes	Yes
Order Routing	Yes	Yes

Supported tags

Tag #	Field Name	Req'd	Comments
	Component Block <Standard Header>	Y	35=1 (MsgType)
112	TestReqID	Y	String to include in the heartbeat message. Data type: String
	Component Block <Standard Trailer>	Y	

Message notes A **Test Request (1)** message is used to force the counterparty to respond with a **Heartbeat (0)** message to validate the state of the connection between a FIX client and the TT FIX Adapter.

When either party stops receiving data for the specified heartbeat interval, it should send a **Test Request (1)** message. The counterparty should send a **Heartbeat (0)** message in response. If a party does not receive a **Heartbeat (0)** message in a reasonable amount of time, the connection should be considered lost.

Related information [Heartbeat \(0\)](#), on page 68

Resend Request (2)

Purpose Used to request a resend of a range of messages

Message direction From TT FIX Adapter to FIX client and from FIX client to TT FIX Adapter

Message availability

Use Case	Price Session	Order Session
Drop Copy	Yes	Yes
Order Routing	Yes	Yes

Supported tags

Tag #	Field Name	Req'd	Comments
	Component Block <Standard Header>	Y	35=2 (MsgType)
7	BeginSeqNo	Y	Sequence number of the first message in the range to resend. Use this value in Tag 34 (MsgSeqNum) of the first resent message. Data type: int
16	EndSeqNo	Y	Sequence number of the last message in the range to resend. Use this value in Tag 34 (MsgSeqNum) of the last message. If the request is for a single message, then Tag 7 (BeginSeqNo) must equal Tag 16 (EndSeqNo). If the request is for all messages subsequent to a particular message, then EndSeqNo must be 0 (representing infinity). Data type: int
	Component Block <Standard Trailer>	Y	

Message notes The **Resend Request (2)** message is used to specify a range of messages to resend. Both the TT FIX Adapter and a FIX client can send this message.

Note: For price sessions, TT FIX Adapter responds to all **Resend Request (2)** messages from FIX clients with a **Sequence Reset (4)** message with Tag 123 (**GapFillFlag**) = Y.

Related information [Logon \(A\)](#), on page 65
[Sequence Reset \(4\)](#), on page 71

Sequence Reset (4)

Purpose Used to reset the message sequence number or gap fill

Message direction From TT FIX Adapter to FIX client and from FIX client to TT FIX Adapter

Message availability

Use Case	Price Session	Order Session
Drop Copy	Yes	Yes
Order Routing	Yes	Yes

Supported tags

Tag #	Field Name	Req'd	Comments
	Component Block <Standard Header>	Y	35=4 (MsgType)
123	GapFillFlag	N	Indicates that the Sequence Reset message replaces missing administrative or application messages. Valid values include: Y : Gap Fill message, Tag 34 (MsgSeqNum) field is valid N : Sequence Reset, ignore Tag 34 (MsgSeqNum) Data type: Boolean Default: N
36	NewSeqNo	Y	New sequence number Data type: SeqNum
	Component Block <Standard Trailer>	Y	

Message notes

The **Sequence Reset (4)** message is used to reset the message sequence number when the FIX client and the TT FIX Adapter message sequence numbers do not match. If the message omits Tag 123 (**GapFillFlag**), the value of Tag 36 (**NewSeqNo**) indicates the new starting sequence number.

If a FIX client or TT FIX Adapter needs to skip messages that would be included in a resend request, it sets Tag 123 (**GapFillFlag**) to Y, and the value of Tag 36 (**NewSeqNo**) represents the sequence number of the next message to be transmitted.

Note: Non-gap fill resets should be used only to when the sequence number differences cannot be resolved.

For more information about sequence resets and gap fills, refer to the FIX 4.2 specification.

Related information

[Logon \(A\)](#), on page 65

[Resend Request \(2\)](#), on page 70

Session-Level Reject (3)

Purpose Used to return the reason for rejecting an incoming message

Message direction From TT FIX Adapter to FIX client and from FIX client to TT FIX Adapter

Message availability

Use Case	Price Session	Order Session
Drop Copy	Yes	Yes
Order Routing	Yes	Yes

Supported tags

Tag #	Field Name	Req'd	Comments
	Component Block <Standard Header>	Y	35=3 (MsgType)
45	RefSeqNum	Y	Reference message sequence number. Data type: Seqnum
58	Text	Y	Reason for rejection. Data type: String
371	RefTagID	Y	Tag number of the referenced FIX message. Data type: int
372	RefMsgType	Y	Message type associated with the referenced FIX message. Data type: String
373	SessionRejectReason	Y	Reason for the rejection. Possible values include: <ul style="list-style-type: none"> 1 : Required tag missing 5 : Incorrect value 6 : Incorrect data format for a tag value 9 : CompID problem Data type: int
	Component Block <Standard Trailer>	Y	

Message notes The **Session-Level Reject (3)** message is used in response to a message that is received but cannot be properly processed due to a session-level rule violation.

Related information Any request

5

Application-Level Messages

Chapter overview

This chapter describes these application-level messages supported by the TT FIX Adapter.

In this chapter

Section	Page
About application-level messages	74
Application message categories	76

About application-level messages

Overview Application-level messages represent business-purpose messages that TT FIX Adapter and FIX clients can exchange with each other. Application-level messages transmit information such as market data, gateway status, orders, and fills.

Supported messages Table 7 describes the application-level FIX messages that TT FIX Adapter currently supports. The Direction column uses the following convention: FC = FIX Client, FA = TT FIX Adapter.

Message	Direction	Description
Business Message Reject (j)	FA to FC	Used to reject a FIX client request for non-session reasons
Execution Report (8)	FA to FC	Used to send order status information to a FIX client, such as confirmations, fills, and unsolicited changes
Gateway Status Request (UAR)	FC to FA	Used to request status information and updates for TT Gateway servers
Gateway Status (UAT)	FA to FC	Used to return status information for TT Gateway servers to FIX clients
Market Data Request (V)	FC to FA	Used to request or manage subscriptions for market data
Market Data Request Reject (Y)	FA to FC	Used to send the reason for rejecting a market data request
Market Data Snapshot/Full Refresh (W)	FA to FC	Used to return a snapshot of market prices
Market Data - Incremental Refresh (X)	FA to FC	Used to send updated market data for an instrument
New Order - Single (D)	FC to FA	Used to submit a new order to TT FIX Adapter
Order Cancel Request (F)	FC to FA	Used to cancel an order
Order Cancel/Replace Request (G)	FC to FA	Used to change an existing order
Order Status Request (H)	FC to FA	Used to request status of orders
Order Cancel Reject (9)	FA to FC	Used to reject an order cancel or cancel/replace request
Request for Position (UAN)	FC to FA	Used to request current positions from the TT FIX Adapter
Position Report (UAP)	FA to FC	Used to return position information
Security Definition Request (c)	FC to FA	Used to request contract information
Security Definition (d)	FA to FC	Used to send contract information in response to a security definition request
Security Status Request (e)	FC to FA	Used to request the status of a product
Security Status (f)	FA to FC	Used to return the current trading status of a security

Table 7. Application-level FIX messages

FIX messages and configuration types Table 8 shows the FIX messages that each type of TT FIX Adapter configuration can exchange with FIX clients

Use case Session type	Drop Copy		Order Routing	
	Price	Order	Price	Order
Business Message Reject (j)	Yes	Yes	Yes	Yes
Execution Report (8)	No	Yes	No	Yes
Gateway Status Request (UAR)	Yes	Yes	Yes	Yes*
Gateway Status (UAT)	Yes	Yes	Yes	Yes*
Market Data Request (V)	Yes	No	Yes	No
Market Data Request Reject (Y)	Yes	No	Yes	No
Market Data Snapshot/Full Refresh (W)	Yes	No	Yes	No
Market Data - Incremental Refresh (X)	Yes	No	Yes	No
New Order - Single (D)	No	No	No	Yes
Order Cancel Request (F)	No	No	No	Yes
Order Cancel/Replace Request (G)	No	No	No	Yes
Order Status Request (H)	No	Yes	No	Yes
Order Cancel Reject (9)	No	No	No	Yes
Request for Position (UAN)	Yes	Yes*	Yes	Yes*
Position Report (UAP)	Yes	Yes*	Yes	Yes*
Security Definition Request (c)	Yes	Yes	Yes	Yes
Security Definition (d)	Yes	Yes	Yes	Yes
Security Status Request (e)	Yes	No	Yes	No
Security Status (f)	Yes	No	Yes	No

* For persistent connections only. If the connection drops, TT FIX Adapter does not deliver responses to subscription requests, nor does it automatically re-subscribe for updates when the connection is restored.

Table 8. Application-level message mappings

Application message categories

Application message categories	<p>TT FIX Adapter supports the following subset of the application messages categories defined in the FIX 4.2 specification:</p> <ul style="list-style-type: none"> • Pre-trade messages • Trade-related messages • Post-trade messages • Other messages
Pre-trade messages	<p>FIX clients use pre-trade messages to gather information that helps determine whether to proceed with trading an instrument. TT FIX Adapter supports the following pre-trade messages:</p> <ul style="list-style-type: none"> • Market Data Request (v) • Market Data Request Reject (y) • Market Data Incremental Refresh (x) • Market Data Snapshot Full Refresh (w) • Security Definition Request (c) • Security Definition (d) • Security Status Request (e) • Security Status (f)
Trade-related messages	<p>TT FIX Adapter and FIX clients use trade-related messages to submit new orders, manage working orders, and report fills. TT FIX Adapter supports the following trade-related messages:</p> <ul style="list-style-type: none"> • Execution Report (8) • New Order Single (D) • Order Cancel Reject (9) • Order Cancel Replace Request (G) • Order Cancel Request (F) • Order Status Request (H)
Post-trade messages	<p>TT FIX Adapter and FIX clients use post-trade messages to communicate position information. TT FIX Adapter supports the following post-trade messages:</p> <ul style="list-style-type: none"> • Position Report (UAP) • Request For Position (UAN)
Other messages	<p>TT FIX Adapter supports the following messages that don't directly involve security trading:</p> <ul style="list-style-type: none"> • Business Message Reject (j) • Gateway Status Request (UAR) • Gateway Status (UAT)

6 Application-Level Messages: Pre-Trade

Chapter overview

This chapter describes pre-trade messages that FIX clients and TT FIX Adapter can exchange during any FIX session.

In this chapter

Section	Page
About Pre-trade Messages	78
Market Data Request (V)	79
Market Data Request Reject (Y)	82
Market Data - Incremental Refresh (X)	89
Market Data Snapshot/Full Refresh (W)	83
Security Definition Request (c)	96
Security Definition (d)	98
Security Status Request (e)	102
Security Status (f)	103

About Pre-trade Messages

Overview Pre-trade messages allow you to retrieve information about securities and market conditions that traders need before deciding whether to submit orders.

Supported messages Table 9 describes the pre-trade FIX messages that TT FIX Adapter currently supports. The Direction column uses the following convention: FC = FIX Client, FA = FIX Adapter.

Message	Direction	Purpose
Market Data Request (V)	FC to FA	Used to request or manage subscriptions for market data
Market Data Request Reject (Y)	FA to FC	Used to send the reason for rejecting a market data request
Market Data - Incremental Refresh (X)	FA to FC	Used to send updated market data for an instrument
Market Data Snapshot/Full Refresh (W)	FA to FC	Used to return a snapshot of market prices
Security Definition Request (c)	FC to FA	Used to request contract information
Security Definition (d)	FA to FC	Used to send contract information in response to a security definition request
Security Status Request (e)	FC to FA	Used to request the status of a product
Security Status (f)	FA to FC	Used to return the current trading status of a security

Table 9. Application messages - pre-trade

FIX messages and configuration types Table 10 shows the FIX messages that each type of TT FIX Adapter configuration can exchange with FIX clients

Use case Session type	Drop Copy		Order Routing	
	Price	Order	Price	Order
Market Data Request (V)	Yes	No	Yes	No
Market Data Request Reject (Y)	Yes	No	Yes	No
Market Data - Incremental Refresh (X)	Yes	No	Yes	No
Market Data Snapshot/Full Refresh (W)	Yes	No	Yes	No
Security Definition Request (c)	Yes	Yes	Yes	Yes
Security Definition (d)	Yes	Yes	Yes	Yes
Security Status Request (e)	Yes	No	Yes	No
Security Status (f)	Yes	No	Yes	No

Table 10. Session-level message mappings

Market Data Request (V)

Purpose Used to request or manage subscriptions for market data

Message direction From FIX client to TT FIX Adapter

Message availability

Use Case	Price Session	Order Session
Drop Copy	Yes	No
Order Routing	Yes	No

Supported tags

Tag #	Field Name	Req'd	Comments
Component Block <Standard Header>		Y	35=V (MsgType)
262	MDReqID	Y	Unique ID for this request. TT FIX Adapter returns this ID in all responses to the request. Data type: String
263	SubscriptionRequestType	Y	Type of request. Valid values include: <ul style="list-style-type: none"> 0: Snapshot 1: Snapshot plus updates (subscribe) 2: Disable previous (unsubscribe) Data type: char
264	MarketDepth	C	Market depth to return in a snapshot. Valid values include: <ul style="list-style-type: none"> 0: Full book 1: Top of book >1: Best specified number of price tiers data Data type: int Condition: Required when Tag 263 (SubscriptionRequestType) is 0 or 1. Note: If this value exceeds the Max Depth Levels setting defined in the FIX Adapter configuration for the FIX session, FIX Adapter returns the depth specified in the configuration. For more information about the setting, refer to the <i>TT FIX Adapter System Administration Manual</i> .
265	MDUpdateType	C	Type of messages TT FIX Adapter sends when updating market data. Valid values include: <ul style="list-style-type: none"> 0: Full refresh 1: Incremental refresh Data type: int Condition: Required when Tag 263 (SubscriptionRequestType) is 1.
266	AggregatedBook	C	Whether the FIX Adapter should send aggregated book entries. Data type: Boolean Condition: Required when Tag 263 (SubscriptionRequestType) is 0 or 1. When used, you must send Y .

Tag #	Field Name	Req'd	Comments
267	NoMDEntryTypes	Y	Number of different Tag 269 (MDEntryType) fields in the request. Data type: NumInGroup Condition: Required when Tag 265 (SubscriptionRequestType) is 0 or 1.
269	MDEntryType	Y	Type of market data to request. Valid values include: 0: Bid 1: Ask 2: Trade 4: Opening price 5: Closing price 6: Settlement price 7: Trading session high price 8: Trading session low price A: Imbalance P: Workup state Y: Implied bid Z: Implied ask n: Market bid o: Market ask p: Indicative open q: Indicative close r: Indicative bid s: Indicative ask t: Indicative settlement u: Local bid v: Local offer w: Dealable regular bid x: Dealable regular offer y: Non-Credit-Screened bid z: Non-Credit-Screened offer Data type: char Condition: Required for the number of requests specified in Tag 267 (NoMDEntryTypes).
146	NoRelatedSym	Y	Number of underlying instruments contained in this repeating group. Data type: NumInGroup
	Component Block <Instrument>	Y	For tags that you include in the Instrument Component Block and any restrictions in their use, refer to the section called Component block: Instrument (FIX client request) , on page 37.
18214	IncludeNumberOfOrders	N	Whether or not to include the number of orders that comprise the quantity in aggregated depth and an indicator as to whether or not that number is exact, when the TT Gateway provides this information. Tag 18214 (IncludeNumberOfOrders) values include Y and N. Data type: char Default Value: N Note: This tag is valid only for market data refresh messages when Tag 269 (MDEntryType) contains: 0 (Bid), 1 (Ask), Y (Implied bid), or Z (Implied ask).
	Component Block <Standard Trailer>	Y	

Message notes

The **Market Data Request (v)** message is used by a FIX client to request market data information from TT FIX Adapter. You can request information like the top of book (Bid, Ask, and session prices) and market depth data.

TT FIX Adapter responds to a **Market Data Request (v)** message with a **Market Data Snapshot Full Refresh (w)** message. Additionally:

- If the FIX Client sets Tag 263 (**SubscriptionRequestType**) to 0, TT FIX Adapter sends no further messages. It does not update the market data.
- If the FIX Client sets Tag 263 (**SubscriptionRequestType**) to 1, TT FIX Adapter updates market data according to the setting the FIX Client uses in Tag 265 (**MDUpdateType**):
 - 0: To update market data, TT FIX Adapter sends **Market Data Snapshot Full Refresh (w)** messages.
 - 1: To update market data, TT FIX Adapter sends **Market Data Incremental Refresh (x)** messages.
- If the FIX Client sets Tag 263 (**SubscriptionRequestType**) to 2, TT FIX Adapter stops sending updates.

If the FIX Adapter cannot fulfill a request for market data because the request contains an invalid instrument, it sends a **Market Data Request Reject (y)** message.

Note: If you submit this request for a TT Gateway whose Price Proxy or Price Server is down, FIX Adapter also responds with a **Market Data Request Reject (y)** message.

Some exchanges disseminate implied prices as well as direct (Bid/Ask) prices. Trying to merge these two separate price streams can result in crossed markets for a brief moment. TT FIX Adapter forwards all market data updates from exchanges, so FIX clients can encounter this scenario as well. FIX clients that subscribe for direct and implied prices should be written to account for this possibility.

Related information

[Market Data Snapshot/Full Refresh \(W\)](#), on page 83

[Market Data Request Reject \(Y\)](#), on page 82

[Market Data - Incremental Refresh \(X\)](#), on page 89

Market Data Request Reject (Y)

Purpose Used to send the reason for rejecting a market data request

Message direction From TT FIX Adapter to FIX client

Message availability

Use Case	Price Session	Order Session
Drop Copy	Yes	No
Order Routing	Yes	No

Supported tags

Tag #	Field Name	Req'd	Comments
	Component Block <Standard Header>	Y	35=Y (MsgType)
262	MDReqID	Y	Unique ID matching the incoming request ID, sent in Tag 262 (MDReqID) in the Market Data Request (v) request. TT FIX Adapter returns this ID in all responses corresponding to the Market Data Request (v) request. Data type: String
58	Text	Y	Reason for the rejection Data type: String
	Component Block <Standard Trailer>	Y	

Message notes The **Market Data Request Reject (Y)** message is used by TT FIX Adapter when it cannot fulfill a FIX client request for market data.

Related information [Market Data Request \(V\)](#), on page 79

Market Data Snapshot/Full Refresh (W)

Purpose Used to return a snapshot of market prices

Message direction From TT FIX Adapter to FIX client

Message availability

Use Case	Price Session	Order Session
Drop Copy	Yes	No
Order Routing	Yes	No

Supported tags

Tag #	Field Name	Req'd	Comments
Component Block <Standard Header>		Y	35=W (MsgType)
262	MDReqID	Y	Unique ID matching the incoming request ID, sent in Tag 262 (MDReqID) in the Market Data Request (v) request. TT FIX Adapter returns this ID in all responses corresponding to the Market Data Request (v) request. Data type: String
Component Block <Instrument>		Y	For tags that you include in the Instrument Component Block and any restrictions in their use, refer to the section called Component block: Instrument (TT FIX Adapter response) , on page 41.
387	TotalVolumeTraded	C	Total volume traded during the current trading session for this instrument. Data type: Qty Condition: Sent if available
18210	PriceFeedStatus	C	Current state of the price feed. Valid values include: 0 : Price feed is unavailable 1 : Price feed is available Data type: int Conditions: Sent only when all of the following are true: <ul style="list-style-type: none"> TT FIX Adapter retrieves the market data from a price proxy environment. The status of the Price Feed changes.

Tag #	Field Name	Req'd	Comments
268	NoMDEntries	Y	Number of market data entries in the message. Starts the repeating group. NumInGroup
Market data entries repeating group			
269	MDEntryType	Y	Type of market data to request. Valid values include: 0 : Bid 1 : Ask 2 : Trade 4 : Opening price 5 : Closing price 6 : Settlement price 7 : Trading session high price 8 : Trading session low price A : Imbalance P : Workup state Y : Implied bid Z : Implied ask n : Market bid o : Market ask p : Indicative open q : Indicative close r : Indicative bid s : Indicative ask t : Indicative settlement u : Local bid v : Local offer w : Dealt regular bid x : Dealt regular offer y : Non-Credit-Screened bid z : Non-Credit-Screened offer Data type: char
270	MDEntryPx	C	Price of the instrument associated with this entry. Interpret the value based on the entry type. Data type: Price Condition: Required unless Tag 269 (MDEntryType)= A (Imbalance), n (Market bid), or o (Market ask).

Tag #	Field Name	Req'd	Comments
271	MDEntrySize	C	<p>Quantity associated with the related instrument.</p> <p>Data type: Qty</p> <p>Condition: Sent when Tag 269 (MDEntryType) contains:</p> <ul style="list-style-type: none"> 0: Bid 1: Offer 2: Trade A: Imbalance Y: Implied bid Z: Implied ask n: Market bid o: Market ask p: Indicative open q: Indicative close r: Indicative bid s: Indicative ask w: Dealable regular bid x: Dealable regular offer <p>Note: Some gateways do not send this tag when Tag 269 (MDEntryType) = 2 (Trade), p (Indicative Open), q (Indicative close), r (Indicative bid), or s (Indicative ask). For more information, refer to the appropriate TT Gateway System Administration Manual.</p>
16487	MDEntrySizeType	C	<p>Type of quantity represented in Tag 271 (MDEntrySize) for the price indicated in Tag 270 (MDEntryPx). Valid values include:</p> <ul style="list-style-type: none"> 1: (Minimum Available) Minimum quantity available at the price level. For example, if Tag 271 (MDEntrySize)=45 and the value of this tag includes 1, at least 45 contracts are available at the price specified in Tag 270 (MDEntryPx). 2: (Net Aggregate) Sum of quantities between the inside market and the price specified in Tag 270 (MDEntryPx). For example, assume Tag 270 (MDEntryPx)=105 and Tag 271 (MDEntrySize)=45. If this tag includes 2, exactly 45 contracts are available between the inside market and the 105 price level. <p>TT FIX Adapter sends this tag when it needs to provide additional information about the quantity specified in Tag 271 (MDEntrySize). When TT FIX Adapter sends this tag, the FIX client should not interpret the value in Tag 271 (MDEntrySize) as a literal value. This tag can also include multiple values to indicate a cumulative effect. For example, with the conditions above, a value in this tag of "1 2" indicates that at least 45 contracts are available between the inside market and the 105 price level.</p> <p>Note: When multiple quantity types apply, the TT FIX Adapter inserts spaces between the values.</p> <p>Data type: String, comprising a space-separated list of values</p> <p>Condition: Sent only when the TT Gateway provides it to the TT FIX Adapter.</p>

Tag #	Field Name	Req'd	Comments
16488	AggressorSide	C	<p>Which party crossed the market to initiate the trade. Valid values include:</p> <ul style="list-style-type: none"> 1: Buyer 2: Seller <p>Data type: int</p> <p>Condition: Sent only when Tag 269 (MDEntryType)=2 (Trade) and the TT Gateway provides the information.</p>
290	MDEntryPositionNo	C	<p>Position of the MD price level in relation to the best bid / best offer (1 being the best).</p> <p>Data type: int</p> <p>Condition: Tag 269 (MDEntryType) is either 0 (bid), 1 (offer), Y (implied bid), or Z (implied offer)</p> <p>Additional Information:</p> <p>Display position of a bid or offer, numbered from most competitive to least competitive, per market side, beginning with 1. The FIX client must determine where the new price belongs based on Tag 270 (MDEntryPx).</p>
198	SecondaryOrderID	C	<p>Unique identifier for the order (assigned by the exchange), based on the following:</p> <ul style="list-style-type: none"> • When Tag 16486 (MDEntryState) = 1 (Open Workup) and Tag 269 (MDEntryType) = 0 (Bid), the value represents the order ID of the best bid at the top of the queue. • When Tag 16486 (MDEntryState) = 1 (Open Workup) and Tag 269 (MDEntryType) = 1 (Offer), the value represents the order ID of the best offer at the top of the queue. • When Tag 16486 (MDEntryState) = (2,4), the value represents the order ID of the workup owner of the ask side. In this situation, the order ID of the work owner on the bid side is the one that was provided in a prior update as the order ID at the top of the queue. (Note: if all offers at the workup price are fully filled, this tag contains the order ID of the best offer at the top of the queue.) • When Tag 16486 (MDEntryState) = (3,5), the value represents the order ID of the workup owner on the bid side. In this situation, the order ID of the workup owner on the ask side is the one that was provided as the order ID of the best ask at the top of the queue. (Note: if all bids at the workup price are fully filled, this tag contains the order ID of the best bid at the top of the queue.) <p>Data type: string</p> <p>Condition: Sent only when Tag 16486 (MDEntryState) is provided.</p> <p>Note: Only the TT BrokerTec Gateway supports this tag.</p>

Tag #	Field Name	Req'd	Comments
16486	MDEntryState	C	<p>Current state of a workup order. Valid values include:</p> <ul style="list-style-type: none"> 1: Open workup 2: Public workup where the aggressor hit the bid 3: Public workup where the aggressor took the ask 4: Private workup where the aggressor hit the bid 5: Private workup where the aggressor took the ask <p>Data type: int Condition: Sent only for workup markets Note: Currently TT FIX Adapter supports BrokerTec workup markets.</p>
277	TradeCondition	C	<p>Whether the quantity represented in Tag 271 (MDEntrySize) represents an imbalance with more buyers or an imbalance with more sellers. Valid values include:</p> <ul style="list-style-type: none"> P: Imbalance More Buyers (Cannot be used in combination with Q) Q: Imbalance More Sellers (Cannot be used in combination with P) <p>Data type: MultipleValueString Condition: Sent when Tag 269 (MDEntryType) = A (Imbalance).</p>
346	NumberOfOrders	C	<p>Number of orders that comprise the quantity represented in Tag 271 (MDEntrySize) of this message.</p> <p>Data type: int</p> <p>Conditions: Sent only when all of the following are true:</p> <ul style="list-style-type: none"> • Tag 18214 (IncludeNumberOfOrders) was set to Y on the Market Data Request (V) message. • Tag 269 (MDEntryType) contains: 0 (Bid), 1 (Ask), Y (Implied bid), or Z (Implied ask). • The TT Gateway provides this information.
16489	ExactNumOrdersIndicator	C	<p>Whether the number of orders specified in Tag 346 (NumberOfOrders) represents an exact or inexact value.</p> <ul style="list-style-type: none"> Y: Exact N: Inexact <p>Data type: Boolean</p> <p>Conditions: Sent only when all of the following are true:</p> <ul style="list-style-type: none"> • Tag 18214 (IncludeNumberOfOrders) was set to Y on the Market Data Request (V) message. • Tag 346 (NumberOfOrders) is provided. • The TT Gateway provides this information.
Component Block <Standard Trailer>		Y	

Message notes	<p>The Market Data Snapshot Full Refresh (w) message is used by TT FIX Adapter to respond to a Market Data Request (v) in the following cases:</p> <ul style="list-style-type: none">• One time, immediately after an incremental subscription before getting the initial incremental updates.• Tag 263 (SubscriptionRequestType) in the request is 0, indicating the client wants a single market snapshot.• Tag 263 (SubscriptionRequestType) in the request is 1 and Tag 265 (MDUpdateType) is 0, indicating the client subscribed to full market updates.
Related information	<p>Market Data - Incremental Refresh (X), on page 89</p> <p>Market Data Request (V), on page 79</p> <p>Market Data Request Reject (Y), on page 82</p>

Market Data - Incremental Refresh (X)

Purpose Used to send updated market data for an instrument

Message direction From TT FIX Adapter to FIX client

Message availability

Use Case	Price Session	Order Session
Drop Copy	Yes	No
Order Routing	Yes	No

Supported tags

Tag #	Field Name	Req'd	Comments
Component Block <Standard Header>		Y	35=X (MsgType)
262	MDReqID	Y	Unique ID matching the incoming request ID, sent in Tag 262 (MDReqID) in the Market Data Request (v) request. TT FIX Adapter returns this ID in all responses corresponding to the Market Data Request (v) request. Data type: String
18210	PriceFeedStatus	C	Current state of the price feed. Valid values include: 0 : Price feed is unavailable 1 : Price feed is available Data type: int Condition: Sent only when TT FIX Adapter retrieves the market data from a price proxy environment.
268	NoMDEntries	Y	Number of market data entries in the message. Starts the repeating group. Data type: NumInGroup
Market data entries repeating group			
279	MDUpdateAction	Y	Type of market data update associated with this repeating group. Valid values include: 0 : New 1 : Change 2 : Delete Data type: char

Tag #	Field Name	Req'd	Comments
269	MDEntryType	Y	<p>Type of market data to request. Valid values include:</p> <ul style="list-style-type: none"> 0: Bid 1: Ask 2: Trade 4: Opening price 5: Closing price 6: Settlement price 7: Trading session high price 8: Trading session low price A: Imbalance P: Workup state Y: Implied bid Z: Implied ask n: Market bid o: Market ask p: Indicative open q: Indicative close r: Indicative bid s: Indicative ask t: Indicative settlement u: Local bid v: Local offer w: Dealable regular bid x: Dealable regular offer y: Non-Credit-Screened bid z: Non-Credit-Screened offer <p>Data type: char</p>
270	MDEntryPx	C	<p>Price of the instrument associated with this entry. Interpret the value based on the entry type.</p> <p>Data type: Price</p> <p>Condition: Required unless Tag 269 (MDEntryType)= A (Imbalance), n (Market bid), or o (Market ask).</p>

Tag #	Field Name	Req'd	Comments
271	MDEntrySize	C	<p>Quantity associated with the related instrument.</p> <p>Data type: Qty</p> <p>Condition: Sent when Tag 269 (MDEntryType) contains:</p> <ul style="list-style-type: none"> 0: Bid 1: Offer 2: Trade A: Imbalance Y: Implied bid Z: Implied ask n: Market bid o: Market ask p: Indicative open q: Indicative close r: Indicative bid s: Indicative ask w: Dealt regular bid x: Dealt regular offer <p>Note: Some gateways do not send this tag when Tag 269 (MDEntryType) = 2 (Trade), p (Indicative open), q (Indicative close), r (Indicative bid), or s (Indicative ask). For more information, refer to the appropriate TT Gateway System Administration Manual.</p>
16487	MDEntrySizeType	C	<p>Type of quantity represented in Tag 271 (MDEntrySize) for the price indicated in Tag 270 (MDEntryPx). Valid values include:</p> <ul style="list-style-type: none"> 1: (Minimum Available) Minimum quantity available at the price level. For example, if Tag 271 (MDEntrySize)=45 and the value of this tag includes 1, at least 45 contracts are available at the price specified in Tag 270 (MDEntryPx). 2: (Net Aggregate) Sum of quantities between the inside market and the price specified in Tag 270 (MDEntryPx). For example, assume Tag 270 (MDEntryPx)=105 and Tag 271 (MDEntrySize)=45. If this tag includes 2, exactly 45 contracts are available between the inside market and the 105 price level. <p>TT FIX Adapter sends this tag when it needs to provide additional information about the quantity specified in Tag 271 (MDEntrySize). When TT FIX Adapter sends this tag, the FIX client should not interpret the value in Tag 271 (MDEntrySize) as a literal value. This tag can also include multiple values to indicate a cumulative effect. For example, with the conditions above, a value in this tag of "1 2" indicates that at least 45 contracts are available between the inside market and the 105 price level.</p> <p>Note: When multiple quantity types apply, the TT FIX Adapter inserts spaces between the values.</p> <p>Data type: String, comprising a space-separated list of values</p> <p>Condition: Sent only when the TT Gateway provides it to the TT FIX Adapter.</p>

Tag #	Field Name	Req'd	Comments
16488	AggressorSide	C	<p>Which party crossed the market to initiate the trade. Valid values include:</p> <ul style="list-style-type: none"> 1: Buyer 2: Seller <p>Data type: int</p> <p>Condition: Sent only when Tag 269 (MDEntryType)=2 (Trade) and the TT Gateway provides the information.</p>
290	MDEntryPositionNo	C	<p>Position of the MD price level in relation to the best bid / best offer (1 being the best). This number always indicates the position of the MD price level before the processing of the current message.</p> <p>Data type: int</p> <p>Condition: Sent when both of the following are true:.</p> <ul style="list-style-type: none"> • Tag 279 (MDUpdateAction) is 1 (change) or 2 (delete) • Tag 269 (MDEntryType) is either 0 (bid), 1 (offer), Y (implied bid), or Z (implied offer) <p>Additional Information:</p> <p>Display position of a bid or offer, numbered from most competitive to least competitive, per market side, beginning with 1. The FIX client must determine where the new price belongs based on Tag 270 (MDEntryPx).</p>
387	TotalVolumeTraded	C	<p>Total volume traded during the current trading session for this instrument.</p> <p>Data type: Qty</p> <p>Condition: Sent if available</p> <p>Note: Included only in the first item in the repeating group</p>
198	SecondaryOrderID	C	<p>Unique identifier for the order (assigned by the exchange), based on the following:</p> <ul style="list-style-type: none"> • When Tag 16486 (MDEntryState) = 1 (Open Workup) and Tag 269 (MDEntryType) = 0 (Bid), the value represents the order ID of the best bid at the top of the queue. • When Tag 16486 (MDEntryState) = 1 (Open Workup) and Tag 269 (MDEntryType) = 1 (Offer), the value represents the order ID of the best offer at the top of the queue. • When Tag 16486 (MDEntryState) = (2,4), the value represents the order ID of the workup owner of the ask side. In this situation, the order ID of the work owner on the bid side is the one that was provided in a prior update as the order ID at the top of the queue. (Note: if all offers at the workup price are fully filled, this tag contains the order ID of the best offer at the top of the queue.) • When Tag 16486 (MDEntryState) = (3,5), the value represents the order ID of the workup owner on the bid side. In this situation, the order ID of the workup owner on the ask side is the one that was provided as the order ID of the best ask at the top of the queue. (Note: if all bids at the workup price are fully filled, this tag contains the order ID of the best bid at the top of the queue.) <p>Data type: string</p> <p>Condition: Sent only when Tag 16486 (MDEntryState) is provided.</p> <p>Note: Only the TT BrokerTec Gateway supports this tag.</p>

Tag #	Field Name	Req'd	Comments
16486	MDEntryState	C	<p>Current state of a workup order. Valid values include:</p> <ul style="list-style-type: none"> 1: Open workup 2: Public workup where the aggressor hit the bid 3: Public workup where the aggressor took the ask 4: Private workup where the aggressor hit the bid 5: Private workup where the aggressor took the ask <p>Data type: int Condition: Sent only for workup markets Note: Currently TT FIX Adapter supports BrokerTec workup markets.</p>
277	TradeCondition	C	<p>Whether the quantity represented in Tag 271 (MDEntrySize) represents an imbalance with more buyers or an imbalance with more sellers. Valid values include:</p> <ul style="list-style-type: none"> P: Imbalance More Buyers (Cannot be used in combination with Q) Q: Imbalance More Sellers (Cannot be used in combination with P) <p>Data type: MultipleValueString Condition: Sent when Tag 269 (MDEntryType) = A (Imbalance).</p>
288	MDEntryBuyer	C	<p>The buying party in the trade.</p> <p>Data type: int Condition: Sent only when Tag 269 (MDEntryType) = 2 (Trade), and when the TT Gateway provides the information. Note: For a list of possible values, refer to the appropriate TT Gateway System Administration Manual.</p>
289	MDEntrySeller	C	<p>The selling party in the trade.</p> <p>Data type: int Condition: Sent only when Tag 269 (MDEntryType) = 2 (Trade), and when the TT Gateway provides the information. Note: For a list of possible values, refer to the appropriate TT Gateway System Administration Manual.</p>
346	NumberOfOrders	C	<p>Number of orders that comprise the quantity represented in Tag 271 (MDEntrySize) of this message.</p> <p>Data type: int Conditions: Sent only when all of the following are true:</p> <ul style="list-style-type: none"> • Tag 18214 (IncludeNumberOfOrders) was set to Y on the Market Data Request (V) message. • Tag 269 (MDEntryType) contains: 0 (Bid), 1 (Ask), Y (Implied bid), or Z (Implied ask). • The TT Gateway provides this information.

Tag #	Field Name	Req'd	Comments
16489	ExactNumOrdersIndicator	C	Whether the number of orders specified in Tag 346 (NumberOfOrders) represents an exact or inexact value. Y : Exact N : Inexact Data type: Boolean Conditions: Sent only when all of the following are true: <ul style="list-style-type: none"> Tag 18214 (IncludeNumberOfOrders) was set to Y on the Market Data Request (V) message. Tag 346 (NumberOfOrders) is provided. The TT Gateway provides this information.
	Component Block <Instrument>	Y	For tags that you include in the Instrument Component Block and any restrictions in their use, refer to the section called Component block: Instrument (TT FIX Adapter response) , on page 41.
	Component Block <Standard Trailer>	Y	

Message notes

The **Market Data Incremental Refresh (x)** message is used to send updates whenever the market data changes for an instrument to which a FIX client subscribes for incremental updates.

Calculating entry positions

Market Data Incremental Refresh (x) messages can contain multiple New, Delete and Change actions for the same contract and MDEntryType. The FIX client must take into account that Tag 290 (**MDEntryPositionNo**) in an update always indicates the position of the entry before processing the current message.

The following shows an example of an algorithm you can use to process market depth updates:

- 1 Create a ladder data structure that is sorted by price.
- 2 Create a before instance and an after instance.
- 3 Populate both instances with the **Market Data Snapshot Full Refresh (w)** message.
- 4 When a **Market Data Incremental Refresh (x)** is received and
 - Tag 279 (**MDUpdateAction**) is 0 (New), insert it directly into the after instance.
 - Tag 279 (**MDUpdateAction**) is 1 (Change), apply the quantity change to the before instance.
 - Tag 279 (**MDUpdateAction**) is 2 (Delete), mark it for deletion in the before instance but do not actually delete it.
- 5 After the entire **Market Data Incremental Refresh (x)** is processed in this manner, insert all records not marked for deletion in the before instance into the after instance.
- 6 Delete the contents of the before instance.
- 7 Copy the contents of after instance to the before instance.
- 8 Display the after instance to the user.

Avoiding duplicate LTP and LTQ updates

When TT FIX Adapter loses its connection to a gateway while a FIX client is connected and subscribed for incremental price updates, the FIX client might receive duplicate LTP/LTQ updates when the connection to the gateway is

reestablished. Thus, TT recommends that FIX clients subscribe for **Gateway Status (UAT)** messages for all gateways from which incremental price updates are being requested. If the FIX client receives a **Gateway Status (UAT)** message indicating that the TT FIX Adapter lost its connection to a gateway, the FIX client should unsubscribe for incremental price updates for all contracts from the indicated gateway. When a **Gateway Status (UAT)** message is received indicating that the connection to the gateway has been reestablished, the FIX client can then re-subscribe.

Related information

[Gateway Status \(UAT\)](#), on page 159

[Market Data Request \(V\)](#), on page 79

[Market Data Snapshot/Full Refresh \(W\)](#), on page 83

[Market Data Request Reject \(Y\)](#), on page 82

Security Definition Request (c)

Purpose Used to request contract information

Message direction From FIX client to TT FIX Adapter

Message availability

Use Case	Price Session	Order Session
Drop Copy	Yes	Yes
Order Routing	Yes	Yes

Supported tags

Tag #	Field Name	Req'd	Comments
	Component Block <Standard Header>	Y	35=c (MsgType)
320	SecurityReqID	Y	Unique ID for this request. TT FIX Adapter includes this tag in its responses to this Security Definition (a) request. Data type: String
321	SecurityRequestType	N	If sent, the FIX Client must set this tag to 3 (Request List Securities) Data type: int
17000	RequestTickTable	N	Whether the FIX Adapter should include a tick table in the response. Valid values include: Y : If available, send a tick table N : Do not send a tick table (default for omitted tag) Data type: String If you need to determine the tick size or tick value for a contract, you must set this value to Y. For more information, see the Message notes section for the Security Definition (a) message.
55	Symbol	N	Exchange-provided product symbol for the tradable product. Data type: String. If you specify a value, TT FIX Adapter returns Security Definition (a) messages for only those contracts whose product symbols match the specified value.
167	SecurityType	N	Asset class of the instrument. Possible values include: CS : common stock FOR : foreign exchange FXNDF : foreign exchange non-deliverable forward FUT : future GOVT : sovereign debt IDX : equity index (valid for market data purposes only) MLEG : multi-leg NRG : energy OPT : option Data type: String If you specify a value, TT FIX Adapter returns Security Definition (a) messages for only those contracts whose security types match the specified value.

Tag #	Field Name	Req'd	Comments
207	SecurityExchange	C	Name of the market where the instrument trades. TT FIX Adapter uses this value to identify a security. Data type: Exchange Condition: Required if Tag 48 (SecurityID) is sent. For a list of supported markets, refer to the section called Tag 207 (SecurityExchange) , on page 206. If you specify a value, TT FIX Adapter returns Security Definition (d) messages for only those contracts whose security exchanges match the specified value.
48	SecurityID	N	TT security ID Data type: String If you specify a value, TT FIX Adapter returns Security Definition (d) messages for only those contracts whose security IDs matches the specified value. Note: If Tag 48 (SecurityID) is provided, you must also provide Tag 207 (SecurityExchange).
Component Block <Standard Trailer>		Y	

Message notes

The **Security Definition Request (c)** message is used by FIX clients to request contract data.

You can filter your security definitions so that TT FIX Adapter supplies only those instruments that match the specified values for one or more of the following tags:

- Tag 55 (Symbol)
- Tag 167 (SecurityType)
- Tag 207 (SecurityExchange)
- Tag 48 (SecurityID)*

* If Tag 48 (SecurityID) is provided, you must also provide Tag 207 (SecurityExchange).

Sending a **Security Definition Request (c)** message instructs TT FIX Adapter to subscribe for contracts that match the filter criteria. TT FIX Adapter sends **Security Definition (d)** messages for all matching contracts and sends new **Security Definition (d)** messages if any of the contract details, such as tick size, change. Additionally, if a TT Gateway adds a new contract that matches the filter criteria, TT FIX Adapter automatically sends a **Security Definition (d)** message for the new contract.

TT FIX Adapter uses AND logic to build the query, so specifying multiple tags reduces the matched securities. If you send none of the tags in the request, TT FIX Adapter returns all securities for all visible TT Gateways.

Note: If TT FIX Adapter finds no contracts that match the specified filter, it sends no response back to the FIX client.

Related information

[Security Definition \(d\)](#), on page 98

Security Definition (d)

Purpose Used to send contract information in response to a security definition request

Message direction From TT FIX Adapter to FIX client

Message availability

Use Case	Price Session	Order Session
Drop Copy	Yes	Yes
Order Routing	Yes	Yes

Supported tags

Tag #	Field Name	Req'd	Comments
Component Block <Standard Header>		Y	35=d (MsgType)
320	SecurityReqID	Y	Unique ID for this request. FIX Adapter populates this tag with the value of the tag in the incoming Security Definition Request (c) message. Data type: String
322	SecurityResponseID	Y	Unique ID for this Security Definition (d) message. Data type: String
323	SecurityResponseType	Y	Type of security definition response. TT FIX Adapter always returns 4, which is the list of securities returned per request. Data type: int
Component Block <Instrument>		C	Condition: Sent unless no securities match the request. For tags included in the Underlying Instrument Component Block and any restrictions in their use, refer to the section called Component block: Instrument (TT FIX Adapter response) , on page 41.
Component Block <Underlying Instrument>		C	Condition: Sent when Tag 167 (SecurityType) is MLEG. For tags included in the Instrument Component Block and any restrictions in their use, refer to the section called Component block: Underlying instrument , on page 41.
393	TotalNumSecurities	Y	Number of securities that match the filter set in the Security Definition Request (c) message. Data type: int
10442	ExpectedFillType	C	Which types of fills FIX Adapter sends for this contract. Possible values include: 1 :Leg and summary fills (sent if the FIX Adapter configuration enables the Send Summary Fills option for the FIX session) 2 :Leg fills only (sent otherwise) Data type: int Condition: Sent if Tag 167 (SecurityType) is MLEG.
16451	PriceDisplayType	Y	Price code indicating how TT formats the price for display purposes in X_TRADER®. Data type: int For a list of price display codes, refer to Price display codes , on page 209.

Tag #	Field Name	Req'd	Comments
16452	TickSize	Y	Size of one tick Data type: int For internal TT use only. Refer to Tag 16552 (ExchTickSize) for the actual tick size.
16454	PointValue	Y	Value of one point in the product's native currency Data type: decimal For internal TT use only. Refer to Tag 16554 (ExchPointValue) for the actual point value.
16456	NumTickTblEntries	Y	Number of ticks in the tick table. Starts the Tick Table repeating group. The message includes this tag and the repeating group when the corresponding Security Definition Request (c) sets Tag 17000 (RequestTickTable) to Y and the contract has a tick table. Data type: NumInGroup See the section called Determining tick sizes and tick values for more information.
16457	NumTicks	C	Tick size multiplier for this price Data type: int Condition: Sent when Tag 16456 (NumTickTblEntries) is greater than 0. See the section called Determining tick sizes and tick values for more information.
16458	MaxPrice	C	Highest price in this price range Data type: Price Condition: Sent when Tag 16456 (NumTickTblEntries) is greater than 0. See the section called Determining tick sizes and tick values for more information.
16461	LotSize	C	Number of delivery days in the contract. Data type: int Condition: Sent only when Tag 167 (SecurityType) is NRG.
16552	ExchTickSize	Y	Size of one base tick for this security Data type: decimal See the section called Determining tick sizes and tick values for more information.
16554	ExchPointValue	Y	Size of one point for this security Data type: decimal See the section called Determining tick sizes and tick values for more information.
16309	ExchContractSymbol	C	The exchange provided contract symbol. Data type: String Condition: Sent when the TT Gateway provides the information.
16899	MarginExcess	C	Margin requirement Data type: decimal Condition: Sent if the Margin parameter is configured in TT User Setup for this product.

Tag #	Field Name	Req'd	Comments
16460	DeliveryUnit	C	Delivery unit for this contract (2500 MMBtus, 50 megawatts, and so on). Sometimes called LotSize. The returned value depends on the value of Tag 16464 (TradesInFlow), as follows: Y :Returns the exchange-provided delivery unit N :Returns 1 Data type: int Condition: Sent only when Tag 167 (SecurityType) is NRG
16463	Blocks	C	Total number of deliverable units per contract Data type: int Condition: Sent only when Tag 167 (SecurityType) is NRG Note: You can calculate the number of deliverable units per delivery day by dividing this value by the value in Tag 16461 (LotSize).
16464	TradesInFlow	C	Whether the contract is a continuously-delivered contract. Possible values include: Y : The contract delivers in flow. N : The contract delivers all-at-once. Data type: character Condition: Sent only when Tag 167 (SecurityType) is NRG
107	SecurityDesc	C	Product description provided by TT Guardian. Data type: String Condition: Sent only if Guardian includes the information in its Description column
18207	TTSecurityAlias	C	Product alias provided by TT Guardian. Data type: String Condition: Sent only if Guardian includes the information in its Alias column
864	NoEvents	Y	Number of events. Starts the Events repeating group. Data type: NumInGroup
Events repeating group			
865	EventType	C	Type of event. Currently, TT FIX Adapter supports only 5 (Expiration Date). Data type: int Condition: Sent when Tag 864 (NoEvents) is greater than 0.
866	EventDate	C	Expiration date Data type: date Condition: Sent when Tag 864 (NoEvents) is greater than 0.
18206	NoGateways	Y	Number of TT Gateways included in the message. Starts the Exchange Gateways repeating group. Data type: int
Gateways repeating group			
18203	ExchangeGateway	C	Name of a specific instance of a TT Gateway, such as CME-A or CME-B. Data type: String Condition: Sent when Tag 18206 (NoGateways) is greater than 0. For more information, refer to Multi-flavor TT Gateway order routing , on page 125
Component Block <Standard Trailer>		Y	

Message notes

The **Security Definition** (d) message is used to send information about an exchange-listed instrument. TT FIX Adapter sends zero or more of these messages to a FIX client in response to a **Security Definition Request** (c). A security definition contains all standing data at the TT Gateways that matches the filter criteria sent in the **Security Definition Request** (c). After the TT FIX Adapter delivers standing data to the FIX client, the TT FIX Adapter sends unsolicited updates to that list of standing data (for instance, if a TT Gateway lists a new product at midday).

Determining tick sizes and tick values

To determine the tick size and tick value of a given contract, the FIX client must:

- In the **Security Definition Request** (c) message, set Tag 17000 (**RequestTickTable**) = Y.
- With the resulting **Security Definition** (d) message, calculate the values using the following algorithm.

```

if tag 16456 (NumTickTblEntries) == 0
{
    (The tick size (and tick value) are the same for all prices.)

    Tick Size (for all prices) = Tag 16552 (ExchTickSize)
    Tick Value (for all prices) =
        Tag 16552 (ExchTickSize) * Tag 16554 (ExchPointValue)
}

else
{
    (The tick size and tick value vary by price level.)

    Base Tick Size = Tag 16552 (ExchTickSize)
    P = price in Points (decimal) for which the tick size is required

    for (int i = 0; i < Tag 16456 (NumTickTblEntries); i++)
    {
        if ( P < (Tag 16458 (MaxPrice) for row i) )
        {
            Tick Size (for price P) =
                Base Tick Size * (Tag 16457 (NumTicks) for row i)

            Tick Value (for price P) =
                Tick Size (for price P) * Tag 16554 (ExchPointValue)

            exit loop
        }
    }
}

```

Working with spreads and strategies

When a FIX client requests a security definition for a spread or strategy, TT FIX Adapter responds by sending:

- A **Security Definition** (d) for the spread itself. An underlying repeating group exists for each leg of the spread. Tag 146 (**NoRelatedSym**) is set to the total number of underlying repeating groups (spread legs) associated with the spread.
- One **Security Definition** (d) for each leg of the spread.

Related information

[Security Definition Request \(c\)](#), on page 96

Security Status Request (e)

Purpose Used to request the status of a product

Message direction From FIX client to TT FIX Adapter

Message availability

Use Case	Price Session	Order Session
Drop Copy	Yes	No
Order Routing	Yes	No

Supported tags

Tag #	Field Name	Req'd	Comments
	Component Block <Standard Header>	Y	35=e (MsgType)
	Component Block <Instrument>	Y	For tags to include in the Instrument Component Block and any restrictions in their use, refer to the section called Component block: Instrument (FIX client request) , on page 37
324	SecurityStatusReqID	Y	Unique ID for this security request. You must ensure the tag contains a unique value for the current FIX session. Data type: String
263	SubscriptionRequestType	Y	Type of request. Valid values include: <ul style="list-style-type: none"> 0: Snapshot 1: Snapshot plus updates (subscribe) 2: Disable previous (unsubscribe) Data type: char
	Component Block <Standard Trailer>	Y	

Message notes The **Security Status Request (e)** message is used by FIX clients to request the current status of a particular product.

Related information [Security Status \(f\)](#), on page 103

Security Status (f)

Purpose Used to return the current trading status of a security

Message direction From TT FIX Adapter to FIX client

Message availability

Use Case	Price Session	Order Session
Drop Copy	Yes	No
Order Routing	Yes	No

Supported tags

Tag #	Field Name	Req'd	Comments
	Component Block <Standard Header>	Y	35=f (MsgType)
324	SecurityStatusReqID	Y	Unique ID sent in the Security Status Request (e) message. FIX Adapter returns this value in all responses associated with the initial request. Data type: String
326	SecurityTradingStatus	Y	Current status of the security. Possible values include: 2 : Trading halted 17 : Ready to trade 18 : Unavailable for trading 19 : Not traded in this market 20 : Unknown or invalid state 21 : Pre-open 23 : Fast market 96 : Auction 97 : Insufficient TTUS permissions to see this contract 98 : Post-trade 99 : Pre-trade Data type: int
58	Text	C	Additional information about the message. Data type: String Condition: Sent if TT FIX Adapter needs to convey additional information.
	Component Block <Instrument>	C	Condition: Sent unless no securities match the request. For tags included in the Instrument Component Block and any restrictions in their use, refer to the section called Component block: Instrument (TT FIX Adapter response) , on page 41.
	Component Block <Standard Trailer>	Y	

Message notes The **Security Status (f)** message is used by TT FIX Adapter to respond to a **Security Status Request (e)** message. The message indicates the current trading status of an instrument.

Related information [Security Status Request \(e\)](#), on page 102

7

Application-Level Messages: Trade-Related

Chapter overview

This chapter describes trade-related messages that FIX clients and TT FIX Adapter can exchange during any FIX session.

In this chapter

	Page
About trade-related messages	106
Execution Report (8)	107
New Order - Single (D)	121
Order Cancel Reject (9)	126
Order Cancel/Replace Request (G)	129
Order Cancel Request (F)	134
Order Status Request (H)	136

About trade-related messages

Overview Trade-related messages allow you to submit, modify, and cancel orders and to receive order acknowledgements.

Supported messages Table 11 describes the trade-related FIX messages that TT FIX Adapter currently supports. The Direction column uses the following convention: FC = FIX Client, FA = FIX Adapter.

Message	Direction	Purpose
Execution Report (8)	FA to FC	Used to send order status information to a FIX client, such as confirmations, fills, and unsolicited changes
New Order - Single (D)	FC to FA	Used to submit a new order to TT FIX Adapter
Order Cancel Reject (9)	FA to FC	Used to reject an order cancel or cancel/replace request
Order Cancel/Replace Request (G)	FC to FA	Used to change an existing order
Order Cancel Request (F)	FC to FA	Used to cancel an order
Order Status Request (H)	FC to FA	Used to request status of orders

Table 11. Application messages - trade-related

FIX messages and configuration types Table 12 shows the FIX messages that each type of TT FIX Adapter configuration can exchange with FIX clients

Use case Session type	Drop Copy		Order Routing	
	Price	Order	Price	Order
Execution Report (8)	No	Yes	No	Yes
New Order - Single (D)	No	No	No	Yes
Order Cancel Reject (9)	No	No	No	Yes
Order Cancel/Replace Request (G)	No	No	No	Yes
Order Cancel Request (F)	No	Yes	No	Yes
Order Status Request (H)	No	No	No	Yes

Table 12. Trade-related messages available per configuration type

Execution Report (8)

Purpose Used to send order status information to a FIX client, such as confirmations, fills, and unsolicited changes

Message direction From TT FIX Adapter to FIX client

Message availability

Use Case	Price Session	Order Session
Drop Copy	No	Yes
Order Routing	No	Yes

Supported tags

Tag #	Field Name	Req'd	Comments
Component Block <Standard Header>		Y	35=8 (MsgType)
17	ExecID	Y	<p>Unique identifier for this execution report. This report ID remains unique until TT FIX Adapter resets.</p> <p>If Tag 150 (ExecType) is 1 (Partial Fill) or 2 (Fill), the value represents a TT-generated fill key containing a maximum of 28 characters.</p> <p>Data type: string</p> <p>Note: Do not try to interpret or parse the value.</p>
19	ExecRefID	C	<p>Reference identifier used with Correct transaction types. Tag 19 (ExecRefID) will be populated with the Tag 17 (ExecID) value of the fill that is being corrected.</p> <p>If there are multiple corrections to a fill, Tag 19 (ExecRefID) will be populated with the Tag 17 (ExecID) value of the most recent execution report for the fill. Tag 16017 (OrigExecID) will always provide the Tag 17 (ExecID) value of the original execution report.</p> <p>Data type: String</p> <p>Condition: Sent when Tag 20 (ExecTransType) = 2 (Correct).</p>
20	ExecTransType	Y	<p>Type of execution report. Possible values include:</p> <ul style="list-style-type: none"> 0: New 2: Correct (when the FIX Adapter configuration enables Send Fill Updates in the FIX Session settings and when either X_TRADER confirms a fill or when an OM Gateway sends a fill update.) 3: Status (in response to an Order Status Request (H) message) <p>Data type: char</p>

Tag #	Field Name	Req'd	Comments
16017	OrigExecID	C	<p>Reference identifier used with Correct transaction types. Tag 16017 (OrigExecID) will be populated with the Tag 17 (ExecID) value of the original fill that has been corrected.</p> <p>If there are multiple corrections to a fill, Tag 16017 (OrigExecID) will always provide the Tag 17 (ExecID) value of the original execution report. Tag 19 (ExecRefID) will be populated with the Tag 17 (ExecID) value of the most recent execution report for the fill.</p> <p>Data type: String</p> <p>Condition: Sent when Tag 20 (ExecTransType) = 2 (Correct)</p>
37	OrderID	Y	<p>Internal TT order key assigned to all orders submitted through any TT software. The value remains constant for the life of an order. In the following situations, the tag contains a value other than an internal key.</p> <p>For messages where the order is unknown, FIX Adapter returns the string NONE.</p> <p>Data type: String</p> <p>Note: The value is unique per TT environment.</p> <p>Additional Information:</p> <p>For more information about using this tag in execution reports, refer to Working with Tag 37 (OrderID), on page 118.</p>
198	SecondaryOrderID	C	<p>Unique identifier for the order, as assigned by the exchange.</p> <p>Data type: String</p> <p>Condition: Sent if available</p> <p>Note: The value can vary over time for the same order.</p>
11	ClOrdID	C	<p>Unique identifier as defined by the client application.</p> <p>If Tag 20 (ExecTransType) is 3 (Status), this message represents a response to an Order Status Request (H) and contains the current Tag 11 (ClOrdID) of the order.</p> <p>Data type: string</p> <p>Conditions:</p> <ul style="list-style-type: none"> Always sent in execution reports for orders placed through the TT FIX Adapter, unless the value has been lost. Not sent for orders that were not created through TT FIX Adapter or in response to Order Status Request (H) that omitted Tag 11 (ClOrdID). TT FIX Adapter sends execution reports without this tag only when the administrator enables either Send Unsolicited Orders or Send Fill Updates in the FIX Session settings within FACT. <p>Note: In non-persistent sessions, the value is lost when a disconnection occurs; in persistent sessions, it is lost when the TT FIX Adapter resets.</p>
21	HandlInst	C	<p>Indicates this report concerns a staged order. TT supports only the value 3 (Staged Order, broker intervention required).</p> <p>Data type: int</p> <p>Condition: Send only for staged orders.</p>

Tag #	Field Name	Req'd	Comments
150	ExecType	Y	<p>This tag indicates the reason that the TT FIX Adapter is sending this execution report. Many other tags in this message depend on the value of this tag. Possible values include:</p> <ul style="list-style-type: none"> 0: New 1: Partial Fill 2: Fill 4: Cancel 5: Replace 6: Pending Cancel 8: Rejected C: Expired D: Restated <p>If Tag 442 (MultiLegReportingType) is 2 (Leg), Tag 150 (ExecType) will be either 1 (Partial Fill) or 2 (Fill). It indicates whether an individual leg of a spread has filled completely.</p> <p>Data type: character</p> <p>Note: Expired statuses are provided if the Send Expired Order Status option is enabled in FACT and when expired order statuses are provided by the TT Gateway.</p>
41	OrigClOrdID	C	<p>Previous order identifier. Equal to Tag 11 (ClOrdID) of the original request message</p> <p>Data type: String</p> <p>Conditions:</p> <ul style="list-style-type: none"> • If Tag 150 (ExecType) is 5 (Replace), this tag contains the value of Tag 11 (ClOrdID) of the order before the replace. This is the Tag 41 (OrigClOrdID) of the Order Cancel Replace Request (G) message that led to this replace confirmation. • If Tag 150 (ExecType) is 4 (Cancel), this tag is present if the 4 (Cancel) is in response to an Order Cancel Request (F) message. In this case, this tag contains the value of Tag 41 (OrigClOrdID) for the cancel message that is being confirmed.
103	OrdRejReason	C	<p>Reason for rejecting the incoming request. Valid values include:</p> <ul style="list-style-type: none"> 0: Broker Option 5: Unknown Order 99: Insufficient TTUS permissions to route orders on the contract 100: Timed Out <p>Data type: int</p> <p>Condition: Sent when Tag 150 (ExecType) is 8 (Reject)</p> <p>Caution: A value of 100 (Timed Out) indicates that New Order Single (D) request received no acknowledgment from the TT Gateway.</p> <p>TT recommends that you contact your risk administrator to determine the status of the order.</p>

Tag #	Field Name	Req'd	Comments
32	LastShares	C	<p>One of the following:</p> <ul style="list-style-type: none"> If Tag 150 (ExecType) is 1 (Partial Fill) or 2 (Fill), this tag indicates the number of contracts that just filled. If Tag 442 (MultiLegReportingType) is 2 (Leg), the value indicates the number of contracts that just filled on the leg of the spread specified by this message's Instrument Component Block. <p>Data type: integer</p> <p>Condition: Sent only when Tag 150 (ExecType) is 1 (Partial Fill) or 2 (Fill)</p>
442	MultiLegReportingType	C	<p>Type of leg reporting for the report. Possible values include:</p> <ol style="list-style-type: none"> 1: Outright 2: Single leg of a multi-leg security 3: Entire multi-leg security (summary fill) <p>Data type: char</p> <p>Conditions: Sent unless Tag 39 (OrdStatus) = 8 (Rejected).</p> <p>Note: TT FIX Adapter sends summary fills (442=3) only when the configuration enables the Send Summary Fills option for FIX sessions. For more information, refer to the <i>TT FIX Adapter System Administration Manual</i>.</p>
39	OrdStatus	Y	<p>Status of the order. Possible values include:</p> <ul style="list-style-type: none"> 0: New 1: Partially filled 2: Filled 4: Canceled 5: Replaced 6: Cancel pending 8: Rejected 9: Suspended (Held) A: Pending new C: Expired E: Pending replace U: Order in an unknown state due to a TT Gateway or Exchange error <p>Note: The trader should immediately contact the Exchange to clarify the status of the order.</p> <p>Data type: char</p> <p>Note: If more than one of these values applies, TT FIX Adapter sends the value required by the FIX 4.2 specification.</p> <p>Note: Expired statuses are provided if the Send Expired Order Status option is enabled in FACT and when expired order statuses are provided by the TT Gateway.</p>
151	LeavesQty	Y	<p>Number of contracts that are still working in the market. Possible values include:</p> <ul style="list-style-type: none"> Tag 38 (OrderQty) - Tag 14 (CumQty), if the order is still in the market 0, otherwise <p>Data type: integer</p> <p>Note: If Tag 442 (MultiLegReportingType) is 2 (Leg), this value represents the total number of contracts that are still working for the leg of the spread specified by this message's Instrument Component Block.</p>

Tag #	Field Name	Req'd	Comments
14	CumQty	Y	Total number of contracts that have filled over the life of this order. It always equals the sum of the Tag 32 (LastShares) of all fill execution reports received for this order, including this one. Data type: integer Note: If Tag 442 (MultiLegReportingType) is 2 (Leg), this value represents the total number of contracts that have filled on the leg of the spread specified by this message's Instrument Component.
58	Text	C	Additional information about the message. Data type: String Condition: Sent if TT FIX Adapter needs to convey additional information.
16728	TotalNumOrders	C	Number of orders included in the response to an order book download request. Data type: integer Condition: Included for order book downloads that do not provide Tag 11 (C1OrdID) or Tag 37 (OrderID). For more information about order book downloads, refer to Order Status Request (H) , on page 136.
NOTE: TT FIX Adapter omits all of the following component blocks and tags for status reject messages.			
Component Block <Instrument>		C	Condition: Sent unless an Order Status Request (H) message returns no orders. For tags included in the Instrument Component Block and any restrictions in their use, refer to the section called Component block: Instrument (TT FIX Adapter response) , on page 41.
Component Block <Underlying Instrument>		C	Condition: Sent when Tag 167 (SecurityType) is MLEG and the TT FIX Adapter configuration enables the Send Security Legs setting for the FIX Session. For tags included in the Underlying Instrument Component Block and any restrictions in their use, refer to the section called Component block: Underlying instrument , on page 44.
Component Block <Trader>		C	Condition: Sent unless an Order Status Request (H) message returns no orders. For tags included in the Trader Component Block and any restrictions in their use, refer to the section called Component block: Trader , on page 49.
44	Price	C	Limit price for the order. Data type: Price Condition: Sent when provided by the TT Gateway and when Tag 40 (OrdType) is: 2 :Limit 4 :Stop Limit 8 :Cross order B :Limit On Close (LOC) O :LSM Q :Market Limit Market (MLM) R :Market to Limit U :Best Limit W :Limit If Touched (LIT)

Tag #	Field Name	Req'd	Comments
99	StopPx	C	<p>Trigger price for a stop order.</p> <p>Data type: Price</p> <p>Condition: Sent when Tag 40 (ordType) is:</p> <ul style="list-style-type: none"> 3: Stop 4: Stop Limit J: Market If Touched (MIT) O: LSM (supported only for TT Eurex and EurexPF Gateways 7.8.0 or higher) S: Stop Market to Limit (without Limit Price) T: Market to Limit (without Limit Price) If-Touched V: Stop Best Limit W: Limit If Touched (LIT) X: Best Limit If Touched (BLIT)
38	OrderQty	C	<p>Total order quantity.</p> <p>Data type: Qty</p> <p>Condition: Sent unless an Order Status Request (H) message returns no orders</p>
110	MinQty	C	<p>Minimum quantity for a Minimum Volume (MV) order.</p> <p>Data type: Qty</p> <p>Condition: Unused for Stop, IOC, or disclosed quantity orders; sent for Minimum Volume (MV) orders</p>
210	MaxShow	C	<p>Quantity disclosed for a disclosed quantity (Iceberg) order.</p> <p>Data type: Qty</p> <p>Condition: Unused for Stop, IOC, or FOK; sent for disclosed quantity (Iceberg) orders</p>
54	Side	C	<p>Side of the order. Possible values include:</p> <ul style="list-style-type: none"> 1: Buy 2: Sell <p>Data type: char</p> <p>Condition: Sent unless an Order Status Request (H) message returns no orders</p>

Tag #	Field Name	Req'd	Comments
40	OrdType	C	<p>Type of order. Possible values include:</p> <ul style="list-style-type: none"> 1:Market 2:Limit 3:Stop 4:Stop Limit 5:Market On Close (MOC) 8:Cross Order B:Limit On Close (LOC) J:Market If Touched (MIT) O:LSM (supported only for TT Eurex and EurexPF Gateways 7.8.0 or higher) Q:Market to Limit (with Limit Price) (MLM) R:Market to Limit (without Limit Price) S:Stop Market to Limit (without Limit Price) T:Market to Limit (without Limit Price) If-Touched U:Best Limit (BL) V:Stop Best Limit W:Limit If Touched (LIT) X:Best Limit If Touched (BLIT) <p>Data type: char</p> <p>Condition: Sent unless an Order Status Request (H) message returns no orders</p> <p>Additional Information:</p> <p>As per the FIX 4.2 specification:</p> <ul style="list-style-type: none"> • For Market On Open (MOO) orders, FIX Adapter sends Tag 40 (OrdType) = 1 (Market) and Tag 59 (TimeInForce) = 2 (At The Opening). • For Limit on Open (LOO) orders, FIX Adapter sends Tag 40 (OrdType) = 2 (Limit) and Tag 59 (TimeInForce) = 2 (At The Opening). <p>TT FIX Adapter supports Limit Stop Market (LSM) orders only for TT Eurex and TT EurexPF Gateways 7.8.0 or higher.</p>
77	OpenClose	C	<p>Whether the order opens or closes a position. Possible values include:</p> <ul style="list-style-type: none"> O:Open (default, if unspecified) C:Close F:FIFO (currently supported only for the TT TOCOM Gateway) <p>When an order is sent with 77=F to TOCOM, the exchange determines whether the order opens or closes a position and sends all corresponding fills with either 77=O or 77=C</p> <p>Data type: char</p> <p>Condition: Sent unless an Order Status Request (H) message returns no orders</p>

Tag #	Field Name	Req'd	Comments
59	TimeInForce	C	How long an order remains active. Possible values include: 0 : Day 1 : Good Till Cancel (GTC) 2 : At The Opening (OPG) 3 : Immediate or Cancel (IOC) 4 : Fill Or Kill (FOK) 6 : Good Till Date 8 : Good in Session (GIS) Z : At The Auction Data type: char Condition: Sent unless an Order Status Request (#) message returns no orders
375	ContraBroker	C	Counterparty data Data type: String Condition: Sent when the TT Gateway provides the information
76	ExecBroker	C	Give up member Data type: String Condition: Sent when provided by the TT Gateway.
432	ExpireDate	C	Date a Good Till Date order expires. Data type: LocalMktDate Condition: Sent when Tag 59 (TimeInForce) = 6 (Good Till Date)
75	TradeDate	C	Exchange session date, in the form: <i>YYYYMMDD</i> . Data type: String Condition: Sent when the TT Gateway provides the information
10541	FixingDate	C	Maturity date, in the form: <i>YYYYMMDD</i> Data type: String Condition: Sent when the TT Gateway provides the information for FXNDF contracts
378	ExecRestatementReason	C	Reason for the restatement. TT FIX Adapter always sends 4 (Broker Option). Data type: integer Condition: Sent when Tag 150 (ExecType) = D (Restated)
16480	PassiveAggressive	C	Whether the order represents a passive or aggressive order. Valid values include: P : Passive A : Aggressive Data type: char Condition: Sent for all BrokerTec orders
16481	AutoAggressive	C	Whether the order auto-aggresses when an opposing passive order attempts to lock the market. Valid values include: Y : Auto-agress N : Do not auto-agress Data type: char Condition: Sent for all BrokerTec orders

Tag #	Field Name	Req'd	Comments
31	LastPx	C	Price for the fill. If Tag 442 (MultiLegReportingType) is 2 (Leg), the value represents the price at which a leg of the spread filled. The message's Instrument Component identifies the filled leg. Data type: decimal Condition: Sent when Tag 150 (ExecType) is 1 (Partial Fill) or 2 (Fill)
6	AvgPx	Y	Average price of all fills on this order, as follows: <ul style="list-style-type: none"> • If order has no fills, TT FIX Adapter sends 0 in this tag. • If the order has fills, TT FIX Adapter sends the sum over all fill execution reports of the product of Tag 32 (LastShares) and Tag 31 (LastPx) divided by the Tag 14 (CumQty). • If Tag 442 (MultiLegReportingType) is 2 (Leg), TT FIX Adapter sends the average price at which the leg filled. Data type: decimal
6038	OrderEnteredTime	C	Time the order was entered or last modified. Data type: UTC timestamp Conditions: <ul style="list-style-type: none"> • Sent only for fills and partial fills. • Sent if provided by the TT Gateway. Additional Information: In certain scenarios, Tag 6038 (OrderEnteredTime) may not be provided in UTC time. For more information about those scenarios, refer to Time Zone of Tag 6038 (OrderEnteredTime) , on page 119.
60	TransactTime	Y	Time, in UTC, the transaction occurred. Data type: UTCTimestamp Note: Some Exchanges might send milliseconds in certain circumstances, in the form: YYYYMMDD-HH:MM:SS.sss. TT recommends that you allocate enough memory to handle the longer timestamp should an Exchange send it.
10527	SecondaryExecID	C	Exchange-generated unique fill ID Data type: String Condition: Sent only for fills (150=1,2) sent by the TT Gateway when available.
16018	TTSessionID	C	The TT Gateway's Session ID. Data type: String Condition: Sent only for fills (150=1,2), when provided by the TT Gateway.
18203	ExchangeGateway	Y	Name of a specific instance of a TT Gateway, such as CME-A or CME-B. Data type: String
277	TradeCondition	C	Whether a BrokerTec order represents a self-trade or that the report was generated by trading against yourself. If TT FIX Adapter sends the tag in an Execution Report (8) message, it always sends it with the value S . Data type: MultipleStringValue Condition: Sent only for BrokerTec self-trade orders

Tag #	Field Name	Req'd	Comments
11028	ManualOrderIndicator	N	Whether the order is sent manually or through automated trading logic. Valid values include: Y : Manual N : Automated Data type: Char Condition: Sent if provided by the TT Gateway
16142	OrderOriginationID	N	Order originator's location Data type: String Condition: Sent if provided by the TT Gateway
16104	TTUserTagData	N	User-defined data supplied with an order. It corresponds to the X_TRADER User Tag field. Data type: String (15 character limit)
16105	TTOrderTagData	N	User-defined data supplied with an order. It corresponds to the X_TRADER Order Tag field. Data type: String (15 character limit)
16106	StagedOrderMsg	C	Message text associated with the staged order Data type: String (256 character limit) Condition: Sent only for staged orders (Tag 21 (HandlInst)=3)
16109	StagedOrderStatus	C	Status of a staged order. Possible values include: A : Available O : Owned Data type: Char Condition: Sent only for staged orders (Tag 21 (HandlInst)=3)
16110	StagedOrderOwner	C	TT Universal Login ID of the broker who took ownership of the staged order. Data type: String Condition: Sent only for staged orders (Tag 21 (HandlInst)=3)
16111	RoutingLevel	C	Indicator of who can work the staged order. Possible values include: B : Broker order visible to traders with TTORD and exchange trader logins I : Internal order visible only to traders with TTORD logins Data type: Character Condition: Sent only for staged orders (Tag 21 (HandlInst)=3)
18208	TTStrategyEngine	C	Name of a specific instance of a TT Strategy Engine, such as SSE-A or SSE-B. Data type: String Condition: Sent only for staged orders (Tag 21 (HandlInst)=3)

Tag #	Field Name	Req'd	Comments
10828	TrdType	N	The type of wholesale trade. Possible values include: 1 : Block Trade 12 : Exchange for Swap (EFS) 94 : Against Actual 95 : Asset Allocation 96 : Basis 97 : Guaranteed Cross 98 : Prof Trade 99 : Vola Data type: int Condition: Sent only for wholesale orders and fills, when provided by the TT Gateway.
18216	ExchCred	C	The exchange order routing credential that was used to route the order. Data type: string Condition: Sent when the TT Gateway provides the information.
136	NoMiscFees	C	Number of miscellaneous fees contained in this repeating group. Data type: NumInGroup; always 1 Condition: Sent when provided by the TT Gateway For more information, see the <i>TT FIX Adapter System Administration Manual</i> .
MiscFees repeating group			
137	MiscFeeAmt	Y	Miscellaneous fee value. Data type: Amt
16112	NoLinks	C	Number of links contained in this repeating group. Data type: int Condition: Sent only when provided by the TT Gateway.
NoLinks repeating group			
16113	LinkID	Y	An identifier used to distinguish that this is a child order or fill that came from a parent order. Child orders and fills that came from the same parent order will have the same LinkID. Data type: string
16114	LinkType	Y	The kind of link. This identifies the relationship between the child orders and fills and the parent order. Valid values include: 1: Spread Quoting 2: Spread Hedge 3: Spread Position Reserve 5: Synthetic Child 6: Synthetic Child Position Reserve 7: Staged Child A: Algo Child U: Unknown Data type: char
Component Block <Standard Trailer>		Y	

Message notes

The **Execution Report (8)** message is used to respond to a successful order request, a fill, or an unsolicited order change.

Handling unsolicited order changes

By default, when an existing order changes from a source other than the originating FIX client session, the **Execution Report (8)** message represents the changed order with Tag 39 (**ordStatus**) set to 5 (Replaced). If you prefer TT FIX Adapter to represent this change with Tag 39 (**ordStatus**) set to 0 (New), you can enable the **Send Unsolicited Change As New** option in the TT FIX Adapter configuration. For more information about this FIX session option, refer to the *TT FIX Adapter System Administration Manual*.

Working with Tag 37 (OrderID)

Tag 37 (**OrderID**) is an internal TT key that is assigned to all orders submitted through any TT software. It keeps the same value for the life of the order:

- if it is cancel/replaced
- if it is partially filled
- if the TT FIX Adapter client connection is terminated or lost
- regardless of how many days an order is working
- regardless of the status of the order
- regardless of how TT FIX Adapter is configured (persistent/non-persistent, etc.)

Tag 37 (**OrderID**) is always unique across client connections and across TT FIX Adapters in the same environment. (The only time that the value is not unique is when a new order is rejected by the TT FIX Adapter. In this case, the value is always "NONE".)

Note: The following situation applies primarily to Drop-Copy TT FIX Adapters.

On rare occasions, TT FIX Adapter might receive a fill, but cannot find the matching order. This situation could occur, for example, if an unsolicited order is both submitted and fully filled while TT FIX Adapter is down, such as during a scheduled or unscheduled reset. When TT FIX Adapter restarts, it automatically downloads data from the Fill and Order Servers. Because the TT Gateway deletes the order from the Order Server when it is fully filled, TT FIX Adapter receives the fill record from the Fill Server but cannot receive the order record because it no longer exists in the Order Server.

The FIX specification requires Tag 37 (**OrderID**) to be unique for all orders. Because TT FIX Adapter cannot determine whether a fill represents a full or partial fill, it creates an artificial, unique value to store in Tag 37 (**OrderID**). Whenever TT FIX Adapter cannot match a fill with its corresponding order, it replaces the value of Tag 37 (**OrderID**) with the concatenation of the values in Tag 37 (**OrderID**) and Tag 17 (**ExecID**).

Order handling differences

TT FIX Adapter processes orders differently than the standard FIX Protocol 4.2, as follows:

- If TT FIX Adapter detects a duplicate Tag 11 (**ClOrdID**) in a **New Order Single (D)**, **Order Cancel Replace Request (G)**, or **Order Cancel Request (F)**, it sends a **Business Message Reject (J)** with Tag 379 (**BusinessRejectRefID**) set to the duplicate ClOrdId. The TT FIX Adapter does not send an **Order Cancel Reject (9)** or **Execution Report (8)**). This prevents programs that monitor order status from becoming confused.
- The TT FIX Adapter does not accept changes to orders if it has not yet processed previous changes to the same order. TT FIX Adapter passes on

cancel requests to the TT Gateway. If an order has a pending action other than cancel, TT FIX Adapter rejects it.

Time Zone of Tag 6038 (OrderEnteredTime)

TT FIX Adapter receives the date and time used to populate Tag 6038 (**OrderEnteredTime**) from fills from TT Gateways. The fills provide the time in the time zone of the TT client application's machine (X_TRADER, XTAPI, Order Routing FIX Adapters, etc). TT FIX Adapter attempts to convert this time to UTC time by applying the difference between the TT FIX Adapter's machine time and UTC time (thus, all orders routed via TT FIX Adapter should have the correct UTC timestamp in Tag 6038 (**OrderEnteredTime**)).

- If the TT client application and the TT FIX Adapter machines share the same time zone, the timestamp provided by TT FIX Adapter in Tag 6038 (**OrderEnteredTime**) should be in UTC time.
- If the TT client application and the TT FIX Adapter machines have different time zones, the timestamp in Tag 6038 (**OrderEnteredTime**) may not be in UTC time. If you prefer to get tag Tag 6038 (**OrderEnteredTime**) values in the TT client application's time zone, set the TT FIX Adapter machine's time zone to UTC time.

Note: If the TT Gateway provides the time in the TT Gateway machine's time zone, the timestamp in Tag 6038 (**OrderEnteredTime**) may not be in UTC time.

Working with the NoLinks Repeating Group

Tag 16114 (**LinkType**) identifies the relationship between the child orders and fills and the parent order.

- Values 1 (Spread Quoting), 2 (Spread Hedge), and 3 (Spread Position Reserve) apply to child orders and fills of Autospreader (whether local X_TRADER Autospreader or Autospreader SE)
- Values 5 (Synthetic Child) and 6 (Synthetic Child Position Reserve) apply to non-staged Synthetic SE child orders and fills.
- Value 7 (Staged Child) applies to staged Synthetic SE child orders and fills.
- Value A (Algo Child) applies to Algo SE child orders and fills.
- Value U (Unknown) applies to any new Link Types that are not yet supported by TT FIX Adapter.

Note: Tag 16113 (**LinkID**) values will be the same for child orders and fills that came from the same parent order.

Related information

[New Order - Single \(D\)](#), on page 121

[Order Cancel Request \(F\)](#), on page 134

[Order Cancel/Replace Request \(G\)](#), on page 129

[Order Status Request \(H\)](#), on page 136

New Order - Single (D)

Purpose Used to submit a new order to TT FIX Adapter

Message direction From FIX client to TT FIX Adapter

Message availability

Use Case	Price Session	Order Session
Drop Copy	No	No
Order Routing	No	Yes

Supported tags

Tag #	Field Name	Req'd	Comments
Component Block <Standard Header>		Y	35=D (MsgType)
11	C1OrdID	Y	Order identifier for the order. This value must be unique since the last FIX Adapter reset. Data type: String
60	TransactTime	C	Time, in UTC, that the FIX Client submitted the message. Data type: UTCTimestamp Condition: Required if the FIX Adapter configuration sets the Stale Order Timeout option for the FIX session in the FIX Adapter configuration. For more information, refer to the <i>TT FIX Adapter System Administration Manual</i> .
Component Block <Instrument>		Y	For tags to include in the Instrument Component Block and any restrictions in their use, refer to the section called Component block: Instrument (FIX client request) , on page 37
Component Block <Underlying Instrument>		Y	Condition: Required when Tag 167 (SecurityType) of the instrument in the Instrument Component Block is MLEG and you specify the instrument by its name instead of its security ID. For more information about specifying instruments by name, refer to the section called Component block: Instrument (FIX client request) , on page 37. For tags to include in the Underlying Instrument Component Block and any restrictions in their use, refer to the section called Component block: Underlying instrument , on page 44.
Component Block <Trader>		Y	For tags to include in the Trader Component Block and any restrictions in their use, refer to the section called Component block: Trader , on page 49.
18	ExecInst	C	Order execution instructions. Possible values include: 1 : Not held (default) 5 : Held Data type: MultipleCharValue To submit a new order in "held" status, set this tag value to 5. Condition: Required when submitting a hold order
21	HandlInst	C	Order handling instructions. TT supports only the value 3 (Staged Order, broker intervention required). Data type: int Condition: Required when submitting a staged order

Tag #	Field Name	Req'd	Comments
44	Price	C	<p>Limit price for limit orders</p> <p>Data type: Price</p> <p>Condition: Required when Tag 40 (ordType) is:</p> <ul style="list-style-type: none"> 2:Limit 4:Stop Limit B:LOC O:LSM (supported only for TT Eurex and EurexPF Gateways 7.8.0 or higher) Q:MLM W:Limit If Touched (LIT) <p>Note: TT FIX Adapter converts the price from a string to a double value. If the converted price is within one ten-thousandth of a tick from a tradable price, TT FIX Adapter rounds the value to the nearest price. Otherwise, TT FIX Adapter rejects the order.</p>
99	StopPx	C	<p>Trigger price for a stop order.</p> <p>Data type: Price</p> <p>Condition: Required when Tag 40 (ordType) is:</p> <ul style="list-style-type: none"> 3:Stop 4:Stop Limit J:Market If Touched (MIT) O:LSM (supported only for TT Eurex and EurexPF Gateways 7.8.0 or higher) S:Stop Market to Limit (without Limit Price) T:Market to Limit (without Limit Price) If-Touched V:Stop Best Limit W:Limit If Touched (LIT) X:Best Limit If Touched (BLIT) <p>Note: TT FIX Adapter converts the price from a string to a double value. If the converted price is within one ten-thousandth of a tick from a tradable price, TT FIX Adapter rounds the value to the nearest price. Otherwise, TT FIX Adapter rejects the order.</p>
38	OrderQty	Y	<p>Total order quantity.</p> <p>Data type: Qty</p>
110	MinQty	C	<p>Minimum quantity for a Minimum Volume (MV) order.</p> <p>Data type: Qty</p> <p>Condition: Required for Minimum Volume (MV) orders</p>
210	MaxShow	C	<p>Quantity to disclose for a disclosed quantity (Iceberg) order.</p> <p>Data type: Qty</p> <p>Condition: Required for disclosed quantity (Iceberg) orders</p>
54	Side	Y	<p>Side of the order. Possible values include:</p> <ul style="list-style-type: none"> 1: Buy 2: Sell <p>Data type: char</p>

Tag #	Field Name	Req'd	Comments
40	OrdType	Y	<p>Type of order. Possible values include:</p> <p>1:Market 2:Limit 3:Stop 4:Stop Limit 5:Market On Close (MOC) B:Limit On Close (LOC) J:Market If Touched (MIT) O:Limit Stop Market (LSM) Q:Market to Limit (with Limit Price) R:Market to Limit (without Limit Price) S:Stop Market to Limit (without Limit Price) T:Market to Limit (without Limit Price) If-Touched U:Best Limit (BL) V:Stop Best Limit W:Limit If Touched (LIT) X:Best Limit If Touched (BLIT)</p> <p>Data type: char</p> <p>Additional Information:</p> <ul style="list-style-type: none"> • Market On Open (MOO) orders require you to set Tag 40 (OrdType) = 1 (Market) and Tag 59 (TimeInForce) = 2 (At The Opening). • Market On Auction (MOA) orders require you to set Tag 40 (OrdType) = 1 (Market) and Tag 59 (TimeInForce) = Z (At the Auction). • Limit on Open (LOO) orders require you to set Tag 40 (OrdType) = 2 (Limit) and Tag 59 (TimeInForce) = 2 (At The Opening). • Market to Limit (without Limit Price) orders do not use Tag 44 (Price). • Market to Limit (without Limit Price) If-Touched orders require you to set a trigger price in Tag 99 (StopPx). • Best Limit (BL) orders do not use Tag 44 (Price). • Best Limit If Touched (BLIT) orders require you to set a trigger price in Tag 99 (StopPx) but not send Tag 44 (Price). • Stop Best Limit orders require you to set a trigger price in Tag 99 (StopPx). • Limit If Touched (LIT) orders require you to set a limit price in Tag 44 (Price) and a trigger price in Tag 99 (StopPx). • Market If Touched (MIT) orders require you to set a trigger price in Tag 99 (StopPx). <p>Notes:</p> <ul style="list-style-type: none"> • Limit Stop Market (LSM) orders are supported only for TT Eurex and TT EurexPF Gateways 7.8.0 or higher. • For staged orders (Tag 21 (HandlInst)=3), FIX Adapter makes this tag value available to the broker, but does not enforce the order type.

Tag #	Field Name	Req'd	Comments
77	OpenClose	N	Whether the order opens or closes a position. Possible values include: O : Open (default, if unspecified) C : Close F : FIFO (currently supported only for the TT TOCOM Gateway) Data type: char
59	TimeInForce	N	How long an order remains active. Possible values include: 0 : Day 1 : Good Till Cancel (GTC) 2 : At The Opening (OPG) 3 : Immediate or Cancel (IOC) 4 : Fill Or Kill (FOK) 6 : Good Till Date 8 : Good in Session (GIS) Z : At The Auction Data type: char; default: 0
432	ExpireDate	C	Date a Good Till Date order expires. Data type: LocalMktDate Condition: Required when Tag 59 (TimeInForce) = 6 (Good Till Date)
11028	ManualOrderIndicator	N	Whether the order is sent manually or through automated trading logic. Valid values include: Y : Manual N : Automated Data type: Char
16142	OrderOriginationID	N	Order originator's location Data type: String Note: TT FIX Adapter forwards this tag only to TT CME and TT CBOT Gateways 7.14.3 and above. Additional Information: If you provide this tag, TT FIX Adapter sends the value to the TT Gateway. Otherwise, TT FIX Adapter tries to populate the value from the information stored in TT User Setup for this FIX client User. If the value still cannot be determined, TT FIX Adapter does not send the tag. For more information about valid values, refer to the TT CME and TT CBOT Gateway System Administration Manuals.
16480	PassiveAggressive	C	Whether to enter the order as a passive or aggressive order. Valid values include: P : Passive A : Aggressive (default if tag omitted) Data type: char Condition: Used only for BrokerTec orders
16481	AutoAggressive	C	Whether the order should auto-aggress when an opposing passive order attempts to lock the market. Valid values include: Y : Auto-agress the order (default if omitted) N : Don't auto-agress the order Data type: char Condition: Used only for BrokerTec orders

Tag #	Field Name	Req'd	Comments
18203	ExchangeGateway	C	Name of a specific instance of a gateway, such as CME-A or CME-B. Data type: String Condition: Required when TT FIX Adapter connects to multiple flavors of the same TT Gateway that list the contract, unless the TT FIX Adapter configuration specifies a Market Name in the Gateway Connection settings that uniquely identifies the specific TT Gateway flavor. For more information, refer to Multi-flavor TT Gateway order routing , on page 125
16104	TTUserTagData	N	Data to supply with an order. It corresponds to the X_TRADER User Tag field. Data type: String (15 character limit)
16105	TTOrderTagData	N	Data to supply with an order. It corresponds to the X_TRADER Order Tag field. Data type: String (15 character limit)
16106	StagedOrderMsg	C	Message text associated with the staged order. Typically used to provide additional information to the broker responsible for managing the order. Data type: String (256 character limit) Condition: Valid only for staged orders (Tag 21 (HandlInst)=3)
16111	RoutingLevel	C	Indicator of who can work the staged order. Possible values include: B : Broker order visible to traders with TTORD and exchange trader logins I : Internal order visible only to traders with TTORD logins Data type: Character; Default: B Condition: Valid only for staged orders (Tag 21 (HandlInst)=3)
Component Block <Standard Trailer>		Y	

Message notes

The **New Order Single (D)** message is used by FIX clients to submit an order for a single contract.

Multi-flavor TT Gateway order routing

TT FIX Adapter supports multi-flavor gateway order routing, as follows:

- If a FIX client user is configured to connect to two (or more) flavors of the same TT Gateway (such as CME-A and CME-B), and the contract for which the order is being routed is available on more than one of these TT Gateways, then this message must set Tag 18203 (**ExchangeGateway**) to indicate to which TT Gateway the order should be routed. Otherwise, Tag 18203 (**ExchangeGateway**) is not required.
- FIX clients can determine the list of TT Gateways on which a contract is available by performing a **Security Definition Request (c)**. The resultant **Security Definition (d)** messages contain this information in tags 18206 (**NoGateways**) and 18203 (**ExchangeGateway**).

Related information

[Execution Report \(8\)](#), on page 107

[Order Cancel/Replace Request \(G\)](#), on page 129

Order Cancel Reject (9)

Purpose Used to reject an order cancel or cancel/replace request

Message direction From TT FIX Adapter to FIX client

Message availability

Use Case	Price Session	Order Session
Drop Copy	No	No
Order Routing	No	Yes

Supported tags

Tag #	Field Name	Req'd	Comments
	Component Block <Standard Header>	Y	35=9 (MsgType)
37	OrderID	Y	Internal TT order key assigned to all orders submitted through any TT software. The value remains constant for the life of an order. In the following situations, the tag contains a value other than an internal key. For messages where the order is unknown, FIX Adapter returns the string NONE . Data type: String
11	c1ordID	Y	Client order identifier specified by the FIX Client; included in Tag 11 (c1ordID) of the initial message Data type: String
41	Origc1ordID	C	Previous order identifier. Equal to Tag 11 (c1ordID) of the original request message Data type: String Condition: Sent if available
198	SecondaryOrderID	C	Unique identifier for the order, as assigned by the exchange. Data type: String
1	Account	C	Order-routing account. The value matches the Tag 1 (Account) in the corresponding request. Data type: String Condition: Sent if the rejected message includes the tag.
10553	TTUsername	C	TT Universal Login ID Data type: String Condition: Sent only if the universal login is configured in TT User Setup and is available.

Tag #	Field Name	Req'd	Comments
39	OrdStatus	Y	<p>Status of the order. Possible values include:</p> <ul style="list-style-type: none"> 0: New 1: Partially filled 2: Filled 4: Cancelled 5: Replaced 6: Cancel pending 8: Rejected 9: Suspended (Held) A: Pending new C: Expired E: Pending replace U: Order in an unknown state due to a TT Gateway or Exchange error <p>Note: The trader should immediately contact the Exchange to clarify the status of the order.</p> <p>Data type: char</p> <p>Note: If more than one of these values applies, TT FIX Adapter sends the value required by the FIX 4.2 specification.</p>
60	TransactTime	Y	<p>Time, in UTC, the transaction occurred.</p> <p>Data type: UTCTimestamp</p> <p>Note: Some Exchanges might send milliseconds in certain circumstances, in the form: <i>YYYYMMDD-HH:MM:SS.sss</i>. TT recommends that you allocate enough memory to handle the longer timestamp should an Exchange send it.</p>
434	CxlRejResponseTo	Y	<p>Type of rejected message. Valid values include:</p> <ul style="list-style-type: none"> 1: Reject is for an Order Cancel Request (F) 2: Reject is for an Order Cancel Replace Request (G) <p>Data type: int</p>
102	CxlRejReason	Y	<p>Reason for rejecting the cancel request. Valid values include:</p> <ul style="list-style-type: none"> 0: Too Late to Cancel 1: Unknown Order 2: Broker Option 3: Action Already Pending 4: Too Early to Cancel (Used only by the TT BrokerTec Gateway) 100: Timed Out <p>Data type: int</p> <p>Caution: A value of 100 (Timed Out) indicates that one of the following order requests received no acknowledgment from the TT Gateway:</p> <ul style="list-style-type: none"> • Order Cancel Request (F) • Order Cancel Replace Request (G) <p>TT recommends that you contact your risk administrator to determine the status of the order.</p>
58	Text	C	<p>Additional information about the message.</p> <p>Data type: String</p> <p>Condition: Sent if TT FIX Adapter needs to convey additional information.</p>
Component Block <Standard Trailer>		Y	

Message notes	The Order Cancel Reject (9) message is used by TT FIX Adapter to reject an Order Cancel Request (F) or Order Cancel Replace Request (G) message.
Related information	Order Cancel/Replace Request (G) , on page 129 Order Cancel Request (F) , on page 134

Order Cancel/Replace Request (G)

Purpose Used to change an existing order

Message direction From FIX client to TT FIX Adapter

Message availability

Use Case	Price Session	Order Session
Drop Copy	No	No
Order Routing	No	Yes

Supported tags

Tag #	Field Name	Req'd	Comments
Component Block <Standard Header>		Y	35=G (MsgType)
41	OrigClOrdID	C	Original order ID. Equal to the current value of Tag 11 (ClOrdID) of the order that this message modifies. Data type: string Condition: Must include either Tag 37 (OrderID) or Tag 41 (OrigClOrdID) Note: Tag 11 (ClOrdID) of an order can change over time.
11	ClOrdID	Y	New ID for the canceled order. This value must be unique since the last TT FIX Adapter reset. Data type: string
37	OrderID	C	Internal TT order key assigned to all orders submitted through any TT software. The value remains constant for the life of an order. Data type: String Condition: Must include either Tag 37 (OrderID) or Tag 41 (OrigClOrdID)
60	TransactTime	C	Time, in UTC, that the FIX Client submitted the message. Data type: UTCTimestamp Condition: Required if the FIX Adapter configuration sets the Stale Order Timeout option for the FIX session in the FIX Adapter configuration. For more information, refer to the <i>TT FIX Adapter System Administration Manual</i> .
Component Block <Instrument>		Y	For tags to include in the Instrument Component Block and any restrictions in their use, refer to the section called Component block: Instrument (FIX client request) , on page 37
Component Block <Underlying Instrument>		Y	Condition: Required when Tag 167 (SecurityType) of the instrument in the Instrument Component Block is MLEG and you specify the instrument by its name instead of its security ID. For more information about specifying instruments by name, refer to the section called Component block: Instrument (FIX client request) , on page 37. For tags to include in the Underlying Instrument Component Block and any restrictions in their use, refer to the section called Component block: Underlying instrument , on page 44.
Component Block <Trader>		Y	For tags to include in the Trader Component Block and any restrictions in their use, refer to the section called Component block: Trader , on page 49.

Tag #	Field Name	Req'd	Comments
18	ExecInst	C	Order execution instructions. Possible values include: 1 : Not held (default) 5 : Held Data type: MultipleCharValue To submit a new order in "held" status, set this tag value to 5. Condition: Required when submitting or releasing a hold order
21	HandlInst	C	Order handling instructions. TT supports only the value 3 (Staged Order, broker intervention required). Data type: int Condition: Required when submitting a staged order
44	Price	C	Limit price for limit orders Data type: Price Condition: Required when Tag 40 (OrdType) is: 2 :Limit 4 :Stop Limit B :LOC O :LSM (supported only for TT Eurex and EurexPF Gateways 7.8.0 or higher) Q :MLM V :Stop Best Limit W :Limit If Touched (LIT) Note : TT FIX Adapter converts the price from a string to a double value. If the converted price is within one ten-thousandth of a tick from a tradable price, TT FIX Adapter rounds the value to the nearest price. Otherwise, TT FIX Adapter rejects the order.
99	StopPx	C	Trigger price for a stop order. Data type: Price Condition: Required when Tag 40 (OrdType) is: 3 :Stop 4 :Stop Limit J :Market If Touched (MIT) O :LSM (supported only for TT Eurex and EurexPF Gateways 7.8.0 or higher) S :Stop Market to Limit (without Limit Price) T :Market to Limit (without Limit Price) If-Touched V :Stop Best Limit W :Limit If Touched (LIT) X :Best Limit If Touched (BLIT) Note : TT FIX Adapter converts the price from a string to a double value. If the converted price is within one ten-thousandth of a tick from a tradable price, TT FIX Adapter rounds the value to the nearest price. Otherwise, TT FIX Adapter rejects the order.
38	OrderQty	Y	Total order quantity. Data type: Qty
110	MinQty	C	Minimum quantity for a Minimum Volume (MV) order. Data type: Qty Condition: Required for Minimum Volume (MV) orders

Tag #	Field Name	Req'd	Comments
210	MaxShow	C	Quantity to disclose for a disclosed quantity (Iceberg) order. Data type: Qty Condition: Required for disclosed quantity (Iceberg) orders
54	Side	Y	Side of the order. Possible values include: 1 : Buy 2 : Sell Data type: char
77	OpenClose	N	Whether the order opens or closes a position. Possible values include: O : Open (default, if unspecified) C : Close F : FIFO (currently supported only for the TT TOCOM Gateway) Data type: char
59	TimeInForce	N	How long an order remains active. Possible values include: 0 : Day 1 : Good Till Cancel (GTC) 2 : At The Opening (OPG) 3 : Immediate or Cancel (IOC) 4 : Fill Or Kill (FOK) 6 : Good Till Date 8 : Good in Session (GIS) Z : At The Auction Data type: char; default: 0
432	ExpireDate	C	Date a Good Till Date order expires. Data type: LocalMktDate Condition: Required when Tag 59 (TimeInForce) = 6 (Good Till Date)
11028	ManualOrderIndicator	N	Whether the order is sent manually or through automated trading logic. Valid values include: Y : Manual N : Automated Data type: Char
16142	OrderOriginationID	N	Order originator's location Data type: String Note: TT FIX Adapter forwards this tag only to TT CME and TT CBOT Gateways 7.14.3 and above. Additional Information: If you provide this tag, TT FIX Adapter sends the value to the TT Gateway. Otherwise, TT FIX Adapter tries to populate the value from the information stored in TT User Setup for this FIX client User. If the value still cannot be determined, TT FIX Adapter does not send the tag. For more information about valid values, refer to the TT CME and TT CBOT Gateway System Administration Manuals.

Tag #	Field Name	Req'd	Comments
40	OrdType	Y	<p>Type of order. Possible values include:</p> <ul style="list-style-type: none"> 1:Market 2:Limit 3:Stop 4:Stop Limit 5:Market On Close (MOC) B:Limit On Close (LOC) J:Market If Touched (MIT) O:Limit Stop Market (LSM) Q:Market to Limit (with Limit Price) R:Market to Limit (without Limit Price) S:Stop Market to Limit (without Limit Price) T:Market to Limit (without Limit Price) If-Touched U:Best Limit (BL) V:Stop Best Limit W:Limit If Touched (LIT) X:Best Limit If Touched (BLIT) <p>Data type: char</p> <p>Additional Information:</p> <ul style="list-style-type: none"> • Market On Open (MOO) orders require you to set Tag 40 (OrdType) = 1 (Market) and Tag 59 (TimeInForce) = 2 (At The Opening). • Market On Auction (MOA) orders require you to set Tag 40 (OrdType) = 1 (Market) and Tag 59 (TimeInForce) = Z (At the Auction). • Limit on Open (LOO) orders require you to set Tag 40 (OrdType) = 2 (Limit) and Tag 59 (TimeInForce) = 2 (At The Opening). • Market to Limit (without Limit Price) orders do not use Tag 44 (Price) • Market to Limit (without Limit Price) If-Touched orders require you to set a trigger price in Tag 99 (StopPx). • Best Limit (BL) orders do not use Tag 44 (Price). • Best Limit If Touched (BLIT) orders require you to set a trigger price in Tag 99 (StopPx) but not send Tag 44 (Price). • Stop Best Limit orders require you to set a trigger price in Tag 99 (StopPx). • Limit If Touched (LIT) orders require you to set a limit price in Tag 44 (Price) and a trigger price in Tag 99 (StopPx). • Market If Touched (MIT) orders require you to set a trigger price in Tag 99 (StopPx). <p>Notes:</p> <ul style="list-style-type: none"> • Limit Stop Market (LSM) orders are supported only for TT Eurex and TT EurexPF Gateways 7.8.0 or higher. • For staged orders (Tag 21 (HandlInst)=3), FIX Adapter makes this tag value available to the broker, but does not enforce the order type.
16480	PassiveAggressive	C	<p>Whether to enter the order as a passive or aggressive order. Valid values include:</p> <ul style="list-style-type: none"> P: Passive A: Aggressive (default if tag omitted) <p>Data type: char</p> <p>Condition: Used only for BrokerTec orders</p>

Tag #	Field Name	Req'd	Comments
16481	AutoAggressive	C	Whether the order should auto-aggress when an opposing passive order attempts to lock the market. Valid values include: Y : Auto-agress the order (default if omitted) N : Don't auto-agress the order Data type: char Condition: Used only for BrokerTec orders
16104	TTUserTagData	N	Data to supply with an order. It corresponds to the X_TRADER User Tag field. Data type: String (15 character limit)
16105	TTOrderTagData	N	Data to supply with an order. It corresponds to the X_TRADER Order Tag field. Data type: String (15 character limit)
16106	StagedOrderMsg	C	Message text associated with the staged order. Typically used to provide additional information to the broker responsible for managing the order. Data type: String (256 character limit) Condition: Valid only for staged orders (Tag 21 (HandlInst)=3)
16111	RoutingLevel	C	Indicator of who can work the staged order. Possible values include: B : Broker order visible to traders with TTORD and exchange trader logins I : Internal order visible only to traders with TTORD logins Data type: Character; Default: B Condition: Valid only for staged orders (Tag 21 (HandlInst)=3)
18203	ExchangeGateway	C	Name of a specific instance of a gateway, such as CME-A or CME-B. Data type: String Condition: Required when TT FIX Adapter connects to multiple flavors of the same TT Gateway that list the contract, unless the TT FIX Adapter configuration specifies a Market Name in the Gateway Connection settings that uniquely identifies the specific TT Gateway flavor.
Component Block <Standard Trailer>		Y	

Message notes

The **Order Cancel Replace Request (G)** message is used by FIX clients to make changes to a previously submitted order.

Related information

[Execution Report \(8\)](#), on page 107
[Order Cancel Reject \(9\)](#), on page 126
[Order Cancel Request \(F\)](#), on page 134

Order Cancel Request (F)

Purpose Used to cancel an order

Message direction From FIX client to TT FIX Adapter

Message availability

Use Case	Price Session	Order Session
Drop Copy	No	No
Order Routing	No	Yes

Supported tags

Tag #	Field Name	Req'd	Comments
Component Block <Standard Header>		Y	35=F (MsgType)
11	C1OrdID	Y	New ID for the canceled order. This value must be unique since the last TT FIX Adapter reset. Data type: string
37	OrderID	C	Internal TT order key assigned to all orders submitted through any TT software. The value remains constant for the life of an order. Data type: String Condition: Must include either Tag 37 (OrderID) or Tag 41 (OrigC1OrdID)
41	OrigC1OrdID	C	Previous order ID. Equal to Tag 11 (C1OrdID) of the original message. Data type: String Condition: Required when the message does not contain Tag 37 (OrderID). Note: Tag 11 (C1OrdID) of an order can change over time. Tag 41 (OrigC1OrdID) must match the current value of Tag 11 (C1OrdID).
11028	ManualOrderIndicator	N	Whether the order is sent manually or through automated trading logic. Valid values include: Y: Manual N: Automated Data type: Char
16142	OrderOriginationID	N	Order originator's location Data type: String Note: TT FIX Adapter forwards this tag only to TT CME and TT CBOT Gateways 7.14.3 and above. Additional Information: If you provide this tag, TT FIX Adapter sends the value to the TT Gateway. Otherwise, TT FIX Adapter tries to populate the value from the information stored in TT User Setup for this FIX client User. If the value still cannot be determined, TT FIX Adapter does not send the tag. For more information about valid values, refer to the TT CME and TT CBOT Gateway System Administration Manuals.
Component Block <Standard Trailer>		Y	

Message notes The **Order Cancel Request** (F) message is used by FIX client to cancel an existing order in the market.

Related information [Order Cancel Reject \(9\)](#), on page 126
[Order Cancel/Replace Request \(G\)](#), on page 129

Order Status Request (H)

Purpose Used to request status of orders

Message direction From FIX client to TT FIX Adapter

Message availability

Use Case	Price Session	Order Session
Drop Copy	No	Yes
Order Routing	No	Yes

Supported tags

Tag #	Field Name	Req'd	Comments
	Component Block <Standard Header>	Y	35=H (MsgType)
1	Account	N	Order-routing account Data type: String You can use this tag to return orders only for a specific account.
11	ClOrdID	N	Order identifier for the order. This value must be unique since the last FIX Adapter reset. Data type: String Note: If you omit Tag 11 and 37, TT FIX Adapter interprets the message as Order Book Download request.
37	OrderID	N	Internal TT order key assigned to all orders submitted through any TT software. The value remains constant for the life of an order. Data type: String Note: If you omit Tag 11 and 37, TT FIX Adapter interprets the message as Order Book Download request.
	Component Block <Standard Trailer>	Y	

Message notes

The **Order Status Request (H)** message is used by FIX clients to query the status of an order. The TT FIX Adapter responds with an **Execution Report (8)** message with Tag 20 (**ExecTransType**) set to 3 (Status) and Tag 150 (ExecType) set to D (Restated).

Downloading order books

If you omit Tag 11 and Tag 37 from an **Order Status Request (H)**, TT FIX Adapter treats the message as a request for all open orders. TT FIX Adapter responds with an **Execution Report (8)** for each working order. Each execution report in response to an order book download request contains Tag 20 (**ExecTransType**) equal to 3 (Status) and Tag 16728 (**TotalNumOrders**) equal to the number of orders sent in the response. For example, when sending 10 orders, TT FIX Adapter sets Tag 16728 (**TotalNumOrders**) to 10 for all 10 responses. When no open orders are available, TT FIX Adapter sends an **Execution Report (8) Reject (39=8, 150=8)**.

Related information

[Execution Report \(8\)](#), on page 107

[Request for Position \(UAN\)](#), on page 139

8 Application-Level Messages: Post-Trade

Chapter overview

This chapter describes post-trade messages that FIX clients and TT FIX Adapter can exchange during any FIX session.

In this chapter

	Page
About post-trade messages	138
Position Report (UAP)	141
Request for Position (UAN)	139

About post-trade messages

Overview Post-trade messages allow you to request and receive position information.

Supported messages Table 1 describes the post-trade FIX messages that TT FIX Adapter currently supports. The Direction column uses the following convention: FC = FIX Client, FA = FIX Adapter.

Message	Direction	Purpose
Position Report (UAP)	FA to FC	Used to return position information
Request for Position (UAN)	FC to FA	Used to request current positions from the TT FIX Adapter

Table 1. Application messages - post-trade

FIX messages and configuration types Table 2 shows the FIX messages that each type of TT FIX Adapter configuration can exchange with FIX clients

Use case Session type	Drop Copy		Order Routing	
	Price	Order	Price	Order
Position Report (UAP)	No	Yes*	No	Yes*
Request for Position (UAN)	No	Yes*	No	Yes*

*For persistent connections only. If the connection drops, TT FIX Adapter does not deliver responses to subscription requests, nor does it automatically re-subscribe for updates when the connection is restored.

Table 2. Post-trade messages available per configuration type

Request for Position (UAN)

Purpose Used to request current positions from the TT FIX Adapter

Message direction From FIX client to TT FIX Adapter

Message availability

Use Case	Price Session	Order Session
Drop Copy	No	Yes
Order Routing	No	Yes

Supported tags

Tag #	Field Name	Req'd	Comments
	Component Block <Standard Header>	Y	35=UAN (MsgType)
1	Account	C	Order-routing account Data type: String Condition: Use when you want to filter the results by account
16710	PosReqId	Y	Unique ID for this request assigned by the FIX Client. Data type: String
16724	PosReqType	Y	Type of position report to request. Valid values include: <ul style="list-style-type: none"> 0: Positions 1: Trades 4: Start of Day (SOD) 5: Manual Fill 6: Detailed Start of Day (DSOD) Data type: int
263	SubscriptionRequestType	N	Type of request. Valid values include: <ul style="list-style-type: none"> 0: Snapshot 1: Snapshot plus updates (subscribe) 2: Disable previous (unsubscribe) Data type: char Default: 0 for positions and fills (16724=0 or 16724=1); 1 for SODs, manual fills, and detailed SODs (16724=4, 16724=5, or 16724=6)
18213	IncludeHistoricalFills	N	Whether to include historical fills in the corresponding position report. Possible values include Y and N. Data type: char; Default: N By default, a UAN message requests fills since the last trading session rollover. When 18213=Y, TT FIX Adapter will send all fills available from the Order Session FIX log files for the past three days. Note: This tag is valid only for trades (16724=1). If you supply the tag for other values of Tag 16724 (PosReqType), TT FIX Adapter rejects the message.
	Component Block <Standard Trailer>	Y	

Message notes The **Request For Position (UAN)** message is used to request:

- Positions

- Trades (historical fills)
- Start-of-Day (SOD) records
- Manual fills
- Detailed Start-of-Day (DSOD) records

TT FIX Adapter responds with a separate **Position Report** (UAP) message for each position, trade, SOD record, manual fill, or DSOD record, based the value of Tag 16724 (**PosReqType**).

Related information

[Execution Report \(8\)](#), on page 107

[Order Status Request \(H\)](#), on page 136

[Position Report \(UAP\)](#), on page 141

Position Report (UAP)

Purpose Used to return position information

Message direction From TT FIX Adapter to FIX client

Message availability

Use Case	Price Session	Order Session
Drop Copy	No	Yes
Order Routing	No	Yes

Supported tags

Tag #	Field Name	Req'd	Comments
	Component Block <Standard Header>	Y	35=UAP (MsgType)
	Component Block <Instrument>	C	Condition: Sent unless no positions match the request. For tags included in the Instrument Component Block and any restrictions in their use, refer to the section called Component block: Instrument (TT FIX Adapter response) , on page 41.
	Component Block <Underlying Instrument>	C	Condition: Sent when Tag 167 (SecurityType) is MLEG and the TT FIX Adapter configuration enables the Send Security Legs option for the FIX session. For tags included in the Underlying Instrument Component Block and any restrictions in their use, refer to the section called Component block: Underlying instrument , on page 44.
	Component Block <Trader>	C	Condition: All tags are conditional, based on the Account Defaults setting in the FIX session for the TT FIX Adapter configuration. For tags included in the Trader Component Block and any restrictions in their use, refer to the section called Component block: Trader , on page 49.
1	Account	C	Order-routing account. The value matches the Tag 1 (Account) in the corresponding request. Data type: String Conditions: <ul style="list-style-type: none"> • Sent if included in the corresponding Request For Position (UAN) request • Always sent when Tag 16727 (TotalNumPosReports) > 0 and Tag 16724 (PosReqType) = 1 (trade), 5 (manual fill), or 6 (detailed SOD) • Never sent when Tag 16724 (PosReqType) = 4 (SOD)

Tag #	Field Name	Req'd	Comments
44	Price	C	<p>Limit price for the order.</p> <p>Data type: Price</p> <p>Condition: Sent when Tag 40 (OrdType) is:</p> <ul style="list-style-type: none"> 2:Limit 4:Stop Limit 8:Cross order B:Limit On Close (LOC) O:LSM Q:Market Limit Market (MLM) R:Market to Limit U:Best Limit W:Limit If Touched (LIT)
99	StopPx	C	<p>Trigger price for a stop order.</p> <p>Data type: Price</p> <p>Condition: Sent when provided by the TT Gateway and when Tag 40 (OrdType) is:</p> <ul style="list-style-type: none"> 3:Stop 4:Stop Limit J:Market If Touched (MIT) O:LSM (supported only for TT Eurex and EurexPF Gateways 7.8.0 or higher) S:Stop Market to Limit (without Limit Price) T:Market to Limit (without Limit Price) If-Touched V:Stop Best Limit W:Limit If Touched (LIT) X:Best Limit If Touched (BLIT)
38	OrderQty	C	<p>Total order quantity.</p> <p>Data type: Qty</p> <p>Condition: Sent when Tag 16727 (TotalNumPosReports) > 0 and Tag 16724 (PosReqType) = 1 (trade) or 5 (manual fill)</p>
110	MinQty	C	<p>Minimum quantity for a Minimum Volume (MV) order.</p> <p>Data type: Qty</p> <p>Condition: Sent for Minimum Volume (MV) orders when Tag 16727 (TotalNumPosReports) > 0 and Tag 16724 (PosReqType) = 1 (trade) or 5 (manual fill)</p>
210	MaxShow	C	<p>Quantity disclosed for a disclosed quantity (Iceberg) order.</p> <p>Data type: Qty</p> <p>Condition: Sent for disclosed quantity (Iceberg) orders when Tag 16727 (TotalNumPosReports) > 0 and Tag 16724 (PosReqType) = 1 (trade) or 5 (manual fill)</p>
54	Side	C	<p>Side of the order. Possible values include:</p> <ul style="list-style-type: none"> 1: Buy 2: Sell <p>Data type: char</p> <p>Condition: Sent when Tag 16727 (TotalNumPosReports) > 0 and Tag 16724 (PosReqType) = 1 (trade), 5 (manual fill), or 6 (detailed SOD)</p>

Tag #	Field Name	Req'd	Comments
40	OrdType	C	<p>Type of order. Possible values include:</p> <ul style="list-style-type: none"> 1:Market 2:Limit 3:Stop 4:Stop Limit 5:Market On Close (MOC) 8:Cross Order B:Limit On Close (LOC) J:Market If Touched (MIT) O:LSM (supported only for TT Eurex and EurexPF Gateways 7.8.0 or higher) Q:Market to Limit (with Limit Price) (MLM) R:Market to Limit (without Limit Price) S:Stop Market to Limit (without Limit Price) T:Market to Limit (without Limit Price) If-Touched U:Best Limit (BL) V:Stop Best Limit W:Limit If Touched (LIT) X:Best Limit If Touched (BLIT) <p>Data type: char</p> <p>Condition: Sent when Tag 16727 (TotalNumPosReports) > 0 and Tag 16724 (PosReqType) = 1 (trade)</p> <p>Additional Information:</p> <p>As per the FIX 4.2 specification:</p> <ul style="list-style-type: none"> • For Market On Open (MOO) orders, TT FIX Adapter sends Tag 40 (OrdType) = 1 (Market) and Tag 59 (TimeInForce) = 2 (At The Opening). • For Limit on Open (LOO) orders, TT FIX Adapter sends Tag 40 (OrdType) = 2 (Limit) and Tag 59 (TimeInForce) = 2 (At The Opening). <p>Note: Limit Stop Market (LSM) orders are supported only for TT Eurex and TT EurexPF Gateways 7.8.0 or higher.</p> <p>Note: For Market On Auction (MOA) orders, the TT FIX Adapter sends Tag 40 (OrdType) = 1 (Market) and Tag 59 (TimeInForce) = Z (At The Auction), even though Market On Auction orders are not part of the FIX 4.2 specification..</p>
77	OpenClose	C	<p>Whether the order opens or closes a position. Possible values include:</p> <ul style="list-style-type: none"> O:Open (default, if unspecified) C:Close F:FIFO (currently supported only for the TT TOCOM Gateway) <p>When an order is sent with 77=F to TOCOM, the exchange determines whether the order opens or closes a position and sends all corresponding fills with either 77=O or 77=C</p> <p>Data type: char</p> <p>Condition: Sent when Tag 16727 (TotalNumPosReports) > 0 and Tag 16724 (PosReqType) = 1 (trade)</p>

Tag #	Field Name	Req'd	Comments
59	TimeInForce	N	How long an order remains active. Possible values include: 0 : Day 1 : Good Till Cancel (GTC) 2 : At The Opening (OPG) 3 : Immediate or Cancel (IOC) 4 : Fill Or Kill (FOK) 6 : Good Till Date 8 : Good in Session (GIS) Z : At The Auction Data type: char Condition: Sent when Tag 16727 (TotalNumPosReports) > 0 and Tag 16724 (PosReqType) = 1 (trade)
375	ContraBroker	C	Counterparty data Data type: String Condition: Sent when the TT Gateway provides the information
76	ExecBroker	C	Give up member Data type: String Condition: Sent when provided by the TT Gateway
432	ExpireDate	C	Date a Good Till Date order expires. Data type: LocalMktDate Conditions: Sent when both of the following are true: <ul style="list-style-type: none"> Tag 59 (TimeInForce) = 6 (Good Till Date) Tag 16727 (TotalNumPosReports) > 0 and Tag 16724 (PosReqType) = 1 (trade) or 5 (manual fill)
75	TradeDate	C	Exchange session date, in the form: <i>YYYYMMDD</i> . Data type: String Conditions: Sent when the TT Gateway provides the information
10541	FixingDate	C	Maturity date, in the form: <i>YYYYMMDD</i> Data type: String Condition: Sent when the TT Gateway provides the information for FXNDF contracts
16480	PassiveAggressive	C	Whether the order represents a passive or aggressive order. Valid values include: P : Passive A : Aggressive Data type: char Condition: Sent for all BrokerTec orders
16481	AutoAggressive	C	Whether the order auto-aggresses when an opposing passive order attempts to lock the market. Valid values include: Y : Auto-agress N : Do not auto-agress Data type: char Condition: Sent for all BrokerTec orders when Tag 16727 (TotalNumPosReports) > 0 and Tag 16724 (PosReqType) = 1 (trade)
16710	PosReqId	Y	Request ID sent in Tag 16710 (PosReqId) of the Request For Position (UAN) request. Data type: String

Tag #	Field Name	Req'd	Comments
16721	PosMaintRptId	Y	Unique ID for this Position Report (UAP) message. Data type: String
16727	TotalNumPosReports	Y	This value indicates the total number of reports expected in response to the Request For Position (UAN) message that generated the report. Note that this value may change in certain scenarios: <ul style="list-style-type: none"> The value will increment as new real-time reports are generated for UAN subscriptions (when Tag 263 (SubscriptionRequestType) = 1 (Snapshot plus updates) in the UAN request). The value can be incremented during initial downloads. A FIX client can determine whether the download has completed by comparing the total number of responses received by the client with the value of Tag 16727 (TotalNumPosReports) in the most recent message. Matching values indicate a completed download. TT FIX Adapter returns 0 if no data exists. Data type: NumInGroup
16724	PosReqType	Y	Type of content in the position report. Valid values include: <ul style="list-style-type: none"> 0: Positions 1: Trades 4: Start of Day (SOD) 5: Manual fills 6: Detailed Start of Day (DSOD) Data type: int
16210	RealizedPandL	C	Realized P&L in native currency for the given instrument. If the TT FIX Adapter configuration defines a rollover time, this value represents the Realized P&L since the last rollover. Data type: decimal Condition: Sent only if Tag 16724 (PosReqType) is 0
32	LastShares	C	One of the following: <ul style="list-style-type: none"> For trades, the number of contracts. For positions, quantity of the open position For SODs, quantity in the SOD For manual fills, quantity in the fill For positions and SODs, positive values indicate long positions and negative values represent short positions. Data type: integer Condition: Not sent when Tag 16727 (TotalNumPosReports) is 0
31	LastPx	C	One of the following: <ul style="list-style-type: none"> 0 (position): average open price 1 (trade): price at which the contracts filled 4 (SOD): price entered in the SOD 5 (manual fill): price of the manual fill 6 (DSOD): price entered in the detailed SOD Data type: decimal Condition: Sent for fills and positions

Tag #	Field Name	Req'd	Comments
37	OrderID	C	<p>Internal TT order key assigned to all orders submitted through any TT software. The value remains constant for the life of an order. In the following situations, the tag contains a value other than an internal key.</p> <p>For messages where the order is unknown, FIX Adapter returns the string NONE.</p> <p>Data type: String</p> <p>Condition: Sent if Tag 16724 (PosReqType) equals 1 (trades) or 5 (manual fills)</p> <p>Additional Information:</p> <p>For more information about using this tag in position reports, refer to Working with Tag 37 (OrderID), on page 153.</p>
198	SecondaryOrderID	C	<p>Unique identifier for the order, as assigned by the exchange.</p> <p>Data type: String</p> <p>Condition: Sent if available when Tag 16724 (PosReqType) = 1 (trade)</p>
17	ExecID	C	<p>Unique trade ID</p> <p>Data type: String</p> <p>Condition: Sent unless Tag 16724 (PosReqType) = 0 (position).</p>
19	ExecRefID	C	<p>Reference identifier used with Correct transaction types. Tag 19 (ExecRefID) will be populated with the Tag 17 (ExecID) value of the fill that is being corrected.</p> <p>If there are multiple corrections to a fill, Tag 19 (ExecRefID) will be populated with the Tag 17 (ExecID) value of the most recent execution report for the fill. Tag 16017 (OrigExecID) will always provide the Tag 17 (ExecID) value of the original execution report.</p> <p>Data type: String</p> <p>Condition: Sent only when Tag 16724 (PosReqType) = 1 (trade), and when Tag 20 (ExecTransType) = 2 (Correct).</p>
20	ExecTransType	C	<p>Type of execution report. Possible values include:</p> <ul style="list-style-type: none"> 0:New 2:Correct (when the FIX Adapter configuration enables Send Fill Updates in the FIX Session settings and when either X_TRADER confirms a fill or the TT BrokerTec Gateway sends a fee update) <p>Data type: char</p> <p>Condition: Sent only when Tag 16724 (PosReqType) = 1 (trade).</p>
16017	OrigExecID	C	<p>Reference identifier used with Correct transaction types. Tag 16017 (OrigExecID) will be populated with the Tag 17 (ExecID) value of the original fill that has been corrected.</p> <p>If there are multiple corrections to a fill, Tag 16017 (OrigExecID) will always provide the Tag 17 (ExecID) value of the original execution report. Tag 19 (ExecRefID) will be populated with the Tag 17 (ExecID) value of the most recent execution report for the fill.</p> <p>Data type: String</p> <p>Condition: Sent only when Tag 16724 (PosReqType) = 1 (trade), and when Tag 20 (ExecTransType) = 2 (Correct)</p>

Tag #	Field Name	Req'd	Comments
58	Text	C	Additional information about the message. Data type: String Condition: Sent if TT FIX Adapter needs to convey additional information.
442	MultiLegReportingType	C	Type of leg reporting for the report. Possible values include: 1: Outright 2: Single leg of a multi-leg security Data type: char Condition: Sent when Tag 16724 (PosReqType) = 1 (trade)
60	TransactTime	C	Time, in UTC, the transaction occurred. Data type: UTCTimestamp Condition: Sent unless Tag 16724 (PosReqType) = 0 (position) Note: Some Exchanges might send milliseconds in certain circumstances, in the form: YYYYMMDD-HH:MM:SS.sss. TT recommends that you allocate enough memory to handle the longer timestamp should an Exchange send it.
10527	SecondaryExecID	C	Exchange-generated unique fill ID Data type: String Condition: Sent only for fills sent by the TT Gateway.
16018	TTSessionID	C	The TT Gateway's Session ID. Data type: String Condition: Sent when provided by the TT Gateway.
18203	ExchangeGateway	C	Name of a specific instance of a TT Gateway, such as CME-A or CME-B. Data type: String Condition: Sent when Tag 16724 (PosReqType) = 1 (trade), 4 (SOD), 5 (manual fill), or 6 (DSOD).
21	HandlInst	C	Indicates this report concerns a staged order. TT supports only the value 3 (Staged Order, broker intervention required). Data type: int Condition: Sent only for staged orders
16104	TTUserTagData	N	User-defined data supplied with an order. It corresponds to the X_TRADER User Tag field. Data type: String (15 character limit)
16105	TTOrderTagData	N	User-defined data supplied with an order. It corresponds to the X_TRADER Order Tag field. Data type: String (15 character limit)
18208	TTStrategyEngine	C	Name of a specific instance of a TT Strategy Engine, such as SSE-A or SSE-B. Data type: String Condition: Sent only for staged orders (Tag 21 (HandlInst)=3)

Tag #	Field Name	Req'd	Comments
10828	TrdType	N	The type of wholesale trade. Possible values include: 1 : Block Trade 12 : Exchange for Swap (EFS) 94 : Against Actual 95 : Asset Allocation 96 : Basis 97 : Guaranteed Cross 98 : Prof Trade 99 : Vola Data type: int Condition: Sent only for wholesale trades, when Tag 16724 (PosReqType) is 1 (Trade), and when provided by the TT Gateway.
18216	ExchCred	C	The exchange order routing credential that was used to route the order. Data type: string Condition: Sent when Tag 16724 (PosReqType) is 1 (Trade), and when provided by the TT Gateway.
136	NoMiscFees	C	Number of miscellaneous fees contained in this repeating group. Data type: NumInGroup; always 1 Condition: Sent when provided by the TT Gateway For more information, see the <i>TT FIX Adapter System Administration Manual</i> .
MiscFees repeating group			
137	MiscFeeAmt	Y	Miscellaneous fee value. Data type: Amt
16112	NoLinks	C	Number of links contained in this repeating group. Data type: int Condition: Sent only if Tag 16724 (PosReqType) is 1, and when provided by the TT Gateway.
NoLinks repeating group			
16113	LinkID	Y	An identifier used to distinguish that this is a child order or fill that came from a parent order. Child orders and fills that came from the same parent order will have the same LinkID. Data type: string
16114	LinkType	Y	The kind of link. This identifies the relationship between the child orders and fills and the parent order. Valid values include: 1: Spread Quoting 2: Spread Hedge 3: Spread Position Reserve 5: Synthetic Child 6: Synthetic Child Position Reserve 7: Staged Child A: Algo Child U: Unknown Data type: char
Component Block <Standard Trailer>		Y	

Message notes

The **Position Report (UAP)** message is used by the TT FIX Adapter to send position information in response to a **Request For Position (UAN)** message from a FIX client. The TT FIX Adapter responds to a single **Request For Position (UAN)** message with a separate **Position Report (UAP)** message for each position, trade, SOD record, manual fill, or DSOD record, based on the value of Tag 16724 (**PosReqType**).

Positions

To request positions, FIX clients send a **Request For Position (UAN)** message with Tag 16724 (**PosReqType**) = 0 (Position). TT FIX Adapter will respond with a separate **Position Report (UAP)** for each contract for which the trader has a position or P&L for only those fills that occurred after the rollover time.

Notes:

- TT FIX Adapter provides only outright positions. It does not send positions for strategies.
- If a FIX client subscribes for positions by including Tag 263 (**SubscriptionRequestType**) = 1 (Snapshot plus updates), the TT FIX Adapter will send a **Position Report (UAP)** as well as an **Execution Report (8)** when the trader gets filled.

In each position report, TT FIX Adapter includes realized P&L in Tag 16210 (**RealizedPandL**) and the average open price in Tag 31 (**LastPx**). For performance reasons, TT FIX Adapter calculates these values using an averaging algorithm.

To help illustrate how TT FIX Adapter uses averaging to calculate realized P&L, the following example walks you through the process TT FIX Adapter uses to process fills. Assume a FIX client receives the set of hypothetical fills for the CBOT:ZB-Mar09 contract shown in Table 3.

Fill	B/S	Qty	Price (Points)
F1	Buy	10	100.5
F2	Sell	8	100.4375
F3	Buy	17	100.34375
F4	Buy	11	100.3125
F5	Sell	9	100.5625

Table 3. P&L calculation: sample fill data

Table 4 shows you how TT FIX Adapter uses the averaging algorithm to calculate unrealized P&L before making it available to FIX clients.

Value	Formula	Result
Total Buy Qty	$F1_{Qty} + F3_{Qty} + F4_{Qty}$ 10 + 17 + 11	38
Avg Buy Price	$((F1_{Qty} * F1_{Price}) + (F3_{Qty} * F3_{Price}) + (F4_{Qty} * F4_{Price})) /$ Total Buy Qty $((10 * 100.5) + (17 * 100.34375) + (11 * 100.3125)) /$ (10 + 17 + 11)	100.3758223
Total Sell Qty	$F2_{Qty} + F5_{Qty}$ 8 + 9	17

Table 4. P&L calculation example: realized P&L

Value	Formula	Result
Avg Sell Price	$((F2_{Qty} * F2_{Price}) + (F5_{Qty} * F5_{Price})) / \text{Total Sell Qty}$ $((8 * 100.4375) + (9 * 100.5625)) / (8 + 9)$	100.5036764
Matched Qty	MINIMUM(Total Buy Qty, Total Sell Qty) MINIMUM(38, 17)	17
Matched Buy Price	Avg Buy Price	100.3758223
Matched Sell Price	Avg Sell Price	100.5036764
Realized P&L (Points)	(Matched Sell Price - Matched Buy Price) * Matched Qty $(100.5036764 - 100.3758223) * 17$	2.1735197
Point Value for CBOT:ZB-Mar09	Value of Tag 16554 (ExchPointValue) in the Security Definition (d) for this contract	\$1000
Realized P&L (Contract Currency (\$))	Realized P&L (Points) * Point Value $2.1735197 * \$1000$	\$2173.52
Unmatched Qty	MAX(Total Buy Qty, Total Sell Qty) - Matched Qty $38 - 17$	21
Avg Open Price	if (Total Buy Qty > Total Sell Qty) Avg Open Price = Avg Buy Price else if (Total Buy Qty < Total Sell Qty) Avg Open Price = Avg Sell Price else Avg Open Price = n/a	100.3758223

Table 4. P&L calculation example: realized P&L(Continued)

TT FIX Adapter automatically calculates realized P&L; it does not, however, provide unrealized P&L. Therefore, if a FIX client wants to determine a trader's total P&L, it must first determine which theoretical exit price to use and calculate the unrealized P&L. Then, the FIX client can plug the value into the following formula:

$$P\&L_{Total} = P\&L_{Realized} + P\&L_{Unrealized}$$

For purposes of this example, assume the theoretical exit price of the position is 100.53125. The FIX client can then calculate the unrealized P&L as shown in Table 5.

Value	Formula	Result
Unrealized P&L (Points)	(Theo Exit Price - Avg Open Price) * Unmatched Qty $(100.53125 - 100.3758223) * 21$	3.2639817
Point Value for CBOT:ZB-Mar09	Value of Tag 16654 (ExchPointValue) in the Security Definition (D) for this contract	\$1000
Unrealized P&L (Contract Currency (\$))	Unrealized P&L (Points) * Point Value $3.2639817 * \$1000$	\$3263.98

Table 5. P&L calculation example: unrealized P&L

With the realized P&L provided by TT FIX Adapter and the newly-calculated P&L, the FIX client can determine the total P&L as shown in Table 6.

Value	Use the following formula	Result
Total P&L (Points)	Realized P&L (Points) + Unrealized P&L (Points) 2.1735197 + 3.2639817	5.4375014
Total P&L (Contract Currency (\$))	Total P&L (Points) * Point Value 5.4375014 * \$1000	\$5437.50

Table 6. P&L calculation example: total P&L

Note: Realized P&L and average open price can be calculated by using a variety of fill matching techniques including FIFO, LIFO, Averaging, etc. Though the values for realized P&L and average open price can vary based on the algorithm chosen, the total P&L is always constant.

TT FIX Adapter client applications that require live P&L should perform a fill download during startup and match fills as shown in the sample scenario. For more information about the averaging technique that TT FIX Adapter uses, refer to Appendix C, [P&L Calculation Algorithm](#)

Trades

To request fills from previous trading sessions, FIX clients send a **Request For Position (UAN)** message with Tag 16724 (**PosReqType**) = 1 (Trades) and Tag 18213 (**IncludeHistoricalFills**) = Y. TT FIX Adapter responds with separate Position Report (UAP) messages for all fills available from the Order Session FIX log files for the past three days. If the **UAN** message omits Tag 18213 or sets its value to N, TT FIX Adapter responds with separate **Position Report (UAP)** messages for only those fills that occurred during the current trading session.

A FIX client can determine whether the fill download completed by comparing the total number of responses received by the client with the value of Tag 16727 (**TotalNumPosReports**) on each message. Matching values indicate a completed fill download.

Manual fills

Risk administrators use manual fills to modify a trader's intra-day position. When a FIX client receives manual fills, it should treat them as regular fills in all position and P&L calculations.

To request manual fills, FIX clients send a **Request For Position (UAN)** message with Tag 16724 (**PosReqType**) = 5 (Manual Fill). TT FIX Adapter responds with separate **Position Report (UAP)** messages for only those manual fills for the current trading session.

Note: By default, Tag 263 (**SubscriptionRequestType**) = 1 (Snapshot plus updates) in **Request For Position (UAN)** messages in which Tag 16724 (**PosReqType**) = 5 (Manual Fill).

SOD records

SOD records are used by risk administrators to roll positions from the prior trading session. A separate SOD record will be injected into the TT system for each contract for which a trader has a position.

Note: The position provided in an SOD record is a summation of positions for all accounts. If account-level position information is required, customers should use DSODs.

Normally, only one SOD record for a contract will be received during the course of a trading session. It should be treated as a regular fill in all position and P&L calculations. If a subsequent SOD record is received for the same contract during the same trading session, FIX clients should replace the original SOD record with the latest one.

To request SOD records, FIX clients send a **Request For Position** (UAN) message with Tag 16724 (PosReqType) = 4 (SOD). When SOD records are injected into the TT system, they always overwrite any existing SOD records. Thus, any SOD records received should always be for the current trading session.

Note: By default, Tag 263 (SubscriptionRequestType) = 1 (Snapshot plus updates) in **Request For Position** (UAN) messages in which Tag 16724 (PosReqType) = 4 (SOD).

You can download SOD records by setting Tag 16724 (PosReqType) to 4 (SOD). If an SOD record contains a NULL price, TT FIX Adapter uses the following rules to determine which price to use when performing P&L and risk calculations:

- Settlement price
- If no settlement price exists, closing price
- If no closing price exists, LTP (last traded price)
- If no LTP exists, price of the matching fill

DSOD records

DSOD records are used by risk administrators to roll positions from the prior trading session. A separate DSOD record will be injected into the TT system for each contract and account for which a trader has a position. When received by FIX clients, they should be treated as a regular fill in all position and P&L calculations.

Note: A separate DSOD record will be injected for all unique combinations of MGT, contract, account, FFT2, and FFT3 values.

To request DSOD records, FIX clients send a **Request For Position** (UAN) message with Tag 16724 (PosReqType) = 6 (DSOD). TT FIX Adapter will respond with a separate **Position Report** (UAP) for only those DSOD records from the current trading session.

Note: By default, Tag 263 (SubscriptionRequestType) = 1 (Snapshot plus updates) in **Request For Position** (UAN) messages in which Tag 16724 (PosReqType) = 6 (DSOD).

If a DSOD record contains a NULL price, TT FIX Adapter uses the following rules to determine which price to use when performing P&L and risk calculations:

- Settlement price
- If no settlement price exists, closing price
- If no closing price exists, LTP (last traded price)
- If no LTP exists, price of the matching fill

Working with Tag 37 (OrderID)

Note: The following situation applies primarily to Drop-Copy TT FIX Adapters.

On rare occasions, TT FIX Adapter might receive a fill, but cannot find the matching order. This situation could occur, for example, if an unsolicited order is both submitted and fully filled while TT FIX Adapter is down, such as during a scheduled or unscheduled reset. When TT FIX Adapter restarts, it automatically downloads data from the Fill and Order Servers. Because the TT Gateway deletes the order from the Order Server when it is fully filled, TT FIX Adapter receives the fill record from the Fill Server but cannot receive the order record because it no longer exists in the Order Server.

The FIX specification requires Tag 37 (**OrderID**) to be unique for all orders. Because TT FIX Adapter cannot determine whether a fill represents a full or partial fill, it creates an artificial, unique value to store in Tag 37 (**OrderID**). Whenever TT FIX Adapter cannot match a fill with its corresponding order, it replaces the value of Tag 37 (**OrderID**) with the concatenation of the values in Tag 37 (**OrderID**) and Tag 17 (**ExecID**).

Working with the NoLinks Repeating Group

Tag 16114 (**LinkType**) identifies the relationship between the child orders and fills and the parent order.

- Values 1 (Spread Quoting), 2 (Spread Hedge), and 3 (Spread Position Reserve) apply to child orders and fills of Autospreader (whether local X_TRADER Autospreader or Autospreader SE)
- Values 5 (Synthetic Child) and 6 (Synthetic Child Position Reserve) apply to non-staged Synthetic SE child orders and fills.
- Value 7 (Staged Child) applies to staged Synthetic SE child orders and fills.
- Value A (Algo Child) applies to Algo SE child orders and fills.
- Value U (Unknown) applies to any new Link Types that are not yet supported by TT FIX Adapter.

Note: Tag 16113 (**LinkID**) values will be the same for child orders and fills that came from the same parent order.

Related information

[Request for Position \(UAN\)](#), on page 139

9

Application-Level Messages: Other

Chapter overview

This chapter describes non-trade-related messages that FIX clients and TT FIX Adapter can exchange during any FIX session.

In this chapter

	Page
About other messages	156
Business Message Reject (j)	157
Gateway Status Request (UAR)	158
Gateway Status (UAT)	159

About other messages

Overview These messages do not directly relate to trading activity.

Supported messages Table 7 describes the non-trade-related FIX messages that TT FIX Adapter currently supports. The Direction column uses the following convention: FC = FIX Client, FA = TT FIX Adapter.

Message	Direction	Purpose
Business Message Reject (j)	FA to FC	Used to reject a FIX client request for non-session reasons
Gateway Status Request (UAR)	FC to FA	Used to request status information and updates for TT Gateway servers
Gateway Status (UAT)	FA to FC	Used to return status information for TT Gateway servers to FIX clients

Table 7. Application messages - other

FIX messages and configuration types Table 8 shows the FIX messages that each type of TT FIX Adapter configuration can exchange with FIX clients

Use case Session type	Drop Copy		Order Routing	
	Price	Order	Price	Order
Business Message Reject (j)	Yes	Yes	Yes	Yes
Gateway Status Request (UAR)	Yes	Yes*	Yes	Yes*
Gateway Status (UAT)	Yes	Yes	Yes	Yes*

*For persistent connections only. If the connection drops, TT FIX Adapter does not deliver responses to subscription requests, nor does it automatically re-subscribe for updates when the connection is restored.

Table 8. Other FIX messages available per configuration type

Business Message Reject (j)

Purpose Used to reject a FIX client request for non-session reasons

Message direction From TT FIX Adapter to FIX client

Message availability

Use Case	Price Session	Order Session
Drop Copy	Yes	Yes
Order Routing	Yes	Yes

Supported tags

Tag #	Field Name	Req'd	Comments
Component Block <Standard Header>		Y	35=j (MsgType)
45	RefSeqNum	Y	Value of Tag 34 (MsgSeqNum) in the rejected message. Data type: integer
372	RefMsgType	Y	Value of Tag 35 (MsgType) in the rejected message. Data type: string
379	BusinessRejectRefID	C	Value of the business-level ID field in the referenced message. Data type: string; Maximum length: 63 characters Condition: Sent when the rejected message contains an ID
380	BusinessRejectReason	Y	Reason for the rejecting the message. Possible values include: 0 : Other 1 : Unknown ID 3 : Unsupported Message Type 5 : Conditionally Required Field Missing Data type: integer
58	Text	C	Text description (free form) of the reason for the rejection. Data type: string Conditions: <ul style="list-style-type: none"> Optional when sent from a FIX client Included in all TT FIX Adapter responses
Component Block <Standard Trailer>		Y	

Message notes The **Business Message Reject (j)** message is used by the TT FIX Adapter or FIX client whenever it rejects an application-level message that fulfills session-level rules but cannot be rejected via any other means (i.e. **Order Cancel Reject (9)** message).

For example, suppose a FIX client sends an **Order Cancel Replace Request (g)** message where Tag 11 (**clOrdID**) matches an existing client order ID. When TT FIX Adapter receives the message, it verifies the validity of the message. However, business rules require that a client order ID remains unique for the trading session. Therefore, TT FIX Adapter responds with a **Business Message Reject (j)** message.

Related information Any application request message

Gateway Status Request (UAR)

Purpose Used to request status information and updates for TT Gateway servers

Message direction From FIX client to TT FIX Adapter

Message availability

Use Case	Price Session	Order Session
Drop Copy	Yes	Yes
Order Routing	Yes	Yes

Supported tags

Tag #	Field Name	Req'd	Comments
	Component Block <Standard Header>	Y	35=UAR (MsgType)
18200	GatewayStatusReqId	Y	Unique ID for this gateway status request. You must ensure the tag contains a unique value for the current FIX session. Data type: String
263	SubscriptionRequestType	Y	Type of request. Valid values include: <ul style="list-style-type: none"> 0: Snapshot 1: Snapshot plus updates (subscribe) 2: Disable previous (unsubscribe) Data type: char
	Component Block <Standard Trailer>	Y	

Message notes The **Gateway Status Request (UAR)** message is used to request the current status snapshots or manage status updates of the following servers on a TT Gateway:

- Price Server
- Order Server
- Fill Server

Related information [Gateway Status \(UAT\)](#), on page 159

Gateway Status (UAT)

Purpose Used to return status information for TT Gateway servers to FIX clients

Message direction From TT FIX Adapter to FIX client

Message availability

Use Case	Price Session	Order Session
Drop Copy	Yes	Yes
Order Routing	Yes	Yes

Supported tags

Tag #	Field Name	Req'd	Comments
Component Block <Standard Header>		Y	35=UAT (MsgType)
18200	GatewayStatusReqId	Y	Unique ID sent in the Gateway Status Request (UAR) message. TT FIX Adapter returns this value in all responses associated with the initial request. Data type: String
325	UnsolicitedIndicator	C	Y if this message is sent unsolicited as a result of a previous subscription request. Data type: Boolean Condition: Sent only if the value is Y Note: The first Gateway Status (UAT) response to a Gateway Status Request (UAR) subscription (tag 263 = 1) will not include Tag 325 (UnsolicitedIndicator), as the first response is reporting the current TT Gateway statuses. Any subsequent Gateway Status (UAT) responses sent as a result of a TT Gateway status change will include Tag 325 (UnsolicitedIndicator) = 'Y' to indicate that a TT Gateway status has changed
18201	NoGatewayStatus	Y	Number of status messages contained in this repeating group Data type: NumInGroup
Gateway status repeating group			
18202	GatewayStatus	Y	Current status of the gateway. Possible values include: 1: Halted 2: Open 3: Closed 4: Pre-Open 5: Pre-Close Data type: int The Pre-Open (4) status indicates that a server is in the process of downloading its data. When the download completes, TT FIX Adapter sends an Open status for the server.
207	SecurityExchange	Y	Name of the exchange (or market) Data type: Exchange For a list of supported markets, refer to the section called Supported TT Gateways , on page 14.

Tag #	Field Name	Req'd	Comments
18203	ExchangeGateway	Y	Name of a specific instance of a TT Gateway, such as CME-A or CME-B. Data type: String
18204	SubExchangeGateway	Y	Type of server on the gateway. Valid values include: <ol style="list-style-type: none"> 1: Price 2: Order 3: Fill Data type: int
58	Text	C	Additional information about the message. Data type: String Condition: Sent if TT FIX Adapter needs to convey additional information.
Component Block <Standard Trailer>		Y	

Message notes

The **Gateway Status (UAT)** message is used to return the status of a TT Gateway in response to a **Gateway Status Request (UAR)** message. The message contains a repeating group including one of the following types of data:

- Snapshots of the current status for each server on each TT Gateway
- Ongoing subscription status changes for each server whose status changes

If FIX Adapter loses its connection to a TT Gateway that is configured as a Price Proxy, it sets the value of Tag 58 (**Text**) to the following:

```
The connection to the price proxy server has been lost. You will be able to receive market data for prior subscriptions, but you will not be able to make new subscriptions until the connection is reestablished.
```

Related information

[Gateway Status Request \(UAR\)](#), on page 158

Chapter overview

This chapter contains sets of request and response conversations between a FIX client and TT FIX Adapter.

In this chapter

Section	Use Case	Page
	Logon and logout	162
	Security download	165
	Security status	171
	Market data	173
	New order	178
	Change order	186
	Positions and fills	190
	Order status	196
	Gateway status	198
	Staged Orders	199

Logon and logout

Overview

These examples show how TT FIX Adapter responds to a **Logon (A)** message in the following scenarios:

- [Sending an invalid SenderCompID](#)
- [Sending an invalid password](#)
- [Logging in with a sequence number lower than expected](#)
- [Logging in with a sequence number greater than expected](#)
- [Logging in with the expected message sequence number](#)
- [Logging in and resetting the message sequence number](#)
- [Logging out](#)

Sending an invalid SenderCompID

The following conversation shows the response when a FIX client sends a **Logon (A)** message with an invalid value for Tag 49 (**SenderCompID**).

Request:

```
8=FIX.4.2+9=84+35=A+34=1+49=REMOTE_INVALID+52=20120309-16:54:02+
56=TT_ORDER+96=12345678+98=0+108=60+141=Y+10=248+
```

Response:

```
8=FIX.4.2+9=00207+35=5+49=TT_ORDER+56=REMOTE_INVALID+34=1+
52=20120309-16:54:02.397+58=Unable to login to adapter Test_FA:
REMOTE_INVALID is not authorized to use FIX Adapter.. Please contact your
System Administrator.+18000=1+10=083+
```

Sending an invalid password

The following conversation shows the response when a FIX client sends a **Logon (A)** message with an invalid password in Tag 96 (**RawData**).

Request:

```
8=FIX.4.2+9=82+35=A+34=1+49=REMOTE+52=20120309-17:37:58+
56=TT_ORDER+96=invalid+98=0+108=60+141=Y+10=083+
```

Response:

```
8=FIX.4.2+9=00215+35=5+49=TT_ORDER+56=REMOTE+34=1+
52=20120309-17:37:59.002+58=Username / Password not recognized.
If you're having trouble logging in, try contacting Trading Systems
Infrastructure tse@tradingtechnologies.com+18000=1+10=160+
```

Logging in with a sequence number lower than expected

In this conversation, a FIX client using a persistent session sends a **Logon (A)** request to TT FIX Adapter with a message sequence number in Tag 34 (**MsgSeqNum**) that is less than TT FIX Adapter expects. TT FIX Adapter responds with a **Logout (5)** message and includes the error message in Tag 58 (**Text**).

Request:

```
8=FIX.4.2+9=93+35=A+49=REMOTE+56=TT_ORDER+34=1+52=20120425-20:55:34.913+
108=15+98=Y+95=8+96=12345678+141=N+10=254+
```

Response :

```
8=FIX.4.2+9=00123+35=5+49=TT_ORDER+56=REMOTE+34=43+
52=20120425-20:55:34.991+58=Sequence number too low. Expected sequence
number: 5+18000=1+10=038+
```

Logging in with a sequence number greater than expected

In this conversation, a FIX client using a persistent session sends a **Logon (A)** request to TT FIX Adapter with a message sequence number in Tag 34 (**MsgSeqNum**) that is greater than TT FIX Adapter expects.

Request :

```
8=FIX.4.2+9=96+35=A+49=REMOTE+56=TT_ORDER+34=2000+
52=20120425-21:37:53.042+108=15+98=Y+95=8+96=12345678+141=N+10=141+
```

TT FIX Adapter responds with two messages. First, it sends a **Logon (A)** message indicating that the client logged on successfully. Then it sends a **Resend Request (2)** message.

Response :

```
8=FIX.4.2+9=00071+35=A+49=TT_ORDER+56=REMOTE+34=78+
52=20120425-21:37:53.062+108=15+98=0+10=069+
```

Request :

```
8=FIX.4.2+9=00070+35=2+49=TT_ORDER+56=REMOTE+34=79+
52=20120425-21:37:53.062+7=703+16=0+10=254+
```

Finally, the FIX client sends the requested message to TT FIX Adapter. In this example, the message happens to be a **Sequence Reset (4)** message.

Response :

```
8=FIX.4.2+9=79+35=4+34=10+43=Y+49=REMOTE+56=TT_ORDER+
52=20120425-21:37:53.153+56=FIX Client+36=17+123=Y+10=048+
```

Logging in with the expected message sequence number

This conversation shows a normal **Logon (A)** message, where a FIX client using a persistent session sends the expected sequence number in Tag 34 (**MsgSeqNum**).

Request :

```
8=FIX.4.2+9=71+35=A+34=21+49=REMOTE+52=20120330-19:23:32+
56=TT_ORDER+96=1+98=0+108=60+10=143+
```

Response :

```
8=FIX.4.2+9=00070+35=A+49=TT_ORDER+56=REMOTE+34=21+
52=20120330-19:23:32.550+108=60+98=0+10=008+
```

Logging in and resetting the message sequence number

In the following conversation, a FIX client using a non-persistent session sends a **Logon (A)** message and sets Tag 141 (**ResetSeqNumFlag**) to reset both the incoming and outgoing message sequence numbers.

Request :

```
FIX.4.2+9=76+35=A+34=1+49=REMOTE+52=20120309-18:51:31+
56=TT_ORDER+96=1234567898+98=0+108=60+141=Y+10=148+
```

Response :

```
8=FIX.4.2+9=00075+35=A+49=TT_ORDER+56=REMOTE+34=1+
52=20120309-18:51:31.541+108=60+141=Y+98=0+10=013+
```

Logging out

This conversations shows a successful logout attempt by a FIX client.

Request :

```
8=FIX.4.2+9=54+35=5+34=20+49=REMOTE+52=20120330-19:23:20+
56=TT_ORDER+10=134+
```

Response :

```
8=FIX.4.2+9=00077+35=5+49=TT_ORDER+56=REMOTE+34=20+
52=20120330-19:23:20.597+58=Received logout+10=122+
```

Security download

Overview

These examples show how TT FIX Adapter responds to a **Security Definition Request (c)** message in the following scenarios:

- [Requesting security definitions](#)
- [Requesting multi-leg security definitions](#)
- [Requesting securities that have tick tables](#)
- [Requesting information for energy securities](#)
- [Requesting a security definition by symbol](#)
- [Requesting a security definition for options](#)
- [Sending an invalid exchange](#)

Requesting security definitions

In this conversation, a FIX client sends a **Security Definition Request (c)** message to request a list of 30-year U.S. Treasury bond (ZB) futures contracts trading on the CBOT exchange. The request also instructs TT FIX Adapter to send any tick tables that exist for these contracts.

Request :

```
8=FIX.4.2+9=95+35=c+34=16+49=REMOTE+52=20120309-20:22:50+
56=TT_PRICE+167=FUT+207=CBOT+55=ZB+320=12345+17000=Y+10=036+
```

TT FIX Adapter responds by sending a **Security Definition (d)** message for each matching security.

Response :

```
8=FIX.4.2+9=00334+35=d+49=TT_PRICE+56=REMOTE+34=18+
52=20120309-20:22:50.125+55=ZB+48=00A0CM00ZBZ+10455=ZBH2+167=FUT+
207=CBOT+15=USD+320=12345+322=12345:0+107=30 Year US Treasury Bond
Futures+18207=bond+200=201203+16451=38+393=3+323=4+16452=0.0078125+
16899=0+16454=1000+16552=0.03125+16554=1000+16456=0+146=0+18206=1+
18203=CBOT+864=1+865=5+866=20120321+10=247+
```

Response :

```
8=FIX.4.2+9=00334++49=TT_PRICE+56=REMOTE+34=19+
52=20120309-20:22:50.125+55=ZB+48=00A0FM00ZBZ+10455=ZBM2+167=FUT+
207=CBOT+15=USD+320=12345+322=12345:1+107=30 Year US Treasury Bond
Futures+18207=bond+200=201206+16451=38+393=3+323=4+16452=0.0078125+
16899=0+16454=1000+16552=0.03125+16554=1000+16456=0+146=0+18206=1+
18203=CBOT+864=1+865=5+866=20120620+10=006+
```

Response :

```
8=FIX.4.2+9=00334+35=d+49=TT_PRICE+56=REMOTE+34=20+
52=20120309-20:22:50.125+55=ZB+48=00A0IM00ZBZ+10455=ZBU2+167=FUT+
207=CBOT+15=USD+320=12345+322=12345:2+107=30 Year US Treasury Bond
Futures+18207=bond+200=201209+16451=38+393=3+323=4+16452=0.0078125+
16899=0+16454=1000+16552=0.03125+16554=1000+16456=0+146=0+18206=1+
18203=CBOT+864=1+865=5+866=20120919+10=024+
```

Requesting multi-leg security definitions

In this conversation, a FIX client sends a **Security Definition Request (c)** message to request a list of DAX spread contracts (FDAX) available on the Eurex exchange, including any tick tables associated with the contracts.

Request :

```
8=FIX.4.2+9=98+35=c+34=9+49=REMOTE+52=20120323-20:47:44+
56=TT_PRICE+167=MLEG+207=Eurex+55=FDAX+17000=Y+320=12345+10=157+
```

TT FIX Adapter responds by sending a **Security Definition (d)** message for each matching contract. Because these contracts are a multi-legged instruments, the response message body contains a repeating group of the underlying leg instruments. In this example, Tag 146 (**NoRelatedSym**) indicates the security contains two leg instruments.

Response :

```
8=FIX.4.2+9=00424+35=d+49=TT_PRICE+56=REMOTE+34=7+
52=20120323-20:47:44.965+55=FDAX+48=FDAX092012SPD122012+10762=Calendar+
167=MLEG+207=Eurex+15=EUR+320=12345+322=12345:0+107=FUT ON DAX INDEX+
16451=0+393=3+10442=1+323=4+16452=0.5+16899=0+16454=25+16552=0.5+
16554=25+16456=0+146=2+311=FDAX+309=FDAX092012+310=FUT+308=Eurex+
318=EUR+313=201209+319=1+54=1+311=FDAX+309=FDAX122012+310=FUT+
308=Eurex+318=EUR+313=201212+319=1+54=2+18206=1+18203=Eurex+10=038+
```

Response :

```
8=FIX.4.2+9=00424+35=d+49=TT_PRICE+56=REMOTE+34=8+
52=20120323-20:47:44.965+55=FDAX+48=FDAX062012SPD092012+10762=Calendar+
167=MLEG+207=Eurex+15=EUR+320=12345+322=12345:1+107=FUT ON DAX INDEX+
16451=0+393=3+10442=1+323=4+16452=0.5+16899=0+16454=25+16552=0.5+
16554=25+16456=0+146=2+311=FDAX+309=FDAX062012+310=FUT+308=Eurex+
318=EUR+313=201206+319=1+54=1+311=FDAX+309=FDAX092012+310=FUT+308=Eurex+
318=EUR+313=201209+319=1+54=2+18206=1+18203=Eurex+10=049+
```

Response :

```
8=FIX.4.2+9=00424+35=d+49=TT_PRICE+56=REMOTE+34=9+
52=20120323-20:47:44.965+55=FDAX+48=FDAX062012SPD122012+10762=Calendar+
167=MLEG+207=Eurex+15=EUR+320=12345+322=12345:2+107=FUT ON DAX INDEX+
16451=0+393=3+10442=1+323=4+16452=0.5+16899=0+16454=25+16552=0.5+
16554=25+16456=0+146=2+311=FDAX+309=FDAX062012+310=FUT+308=Eurex+
318=EUR+313=201206+319=1+54=1+311=FDAX+309=FDAX122012+310=FUT+308=Eurex+
318=EUR+313=201212+319=1+54=2+18206=1+18203=Eurex+10=033+
```

Requesting securities that have tick tables

In this conversation, a FIX client sends a **Security Definition Request (c)** message to request a list of FDAX futures trading on the Eurex exchange. In the message body, the client also includes Tag 17000 (**RequestTickTable**) to request tick table information for the security.

Request :

```
8=FIX.4.2+9=97+35=c+34=7+49=REMOTE+52=20120403-20:43:01+56=TT_ORDER+
167=FUT+207=Eurex+55=FDAX+320=12345+17000=Y+10=097+
```

TT FIX Adapter responds by sending a **Security Definition (d)** message for each matching security. Because FDAX futures provide tick tables, the response message body contains a repeating group of them. In this example, Tag 16456 (**NumTickTblEntries**) indicates the security contains one tick table element.

Response :

```
8=FIX.4.2+9=00285+35=d+49=TT_ORDER+56=REMOTE+34=25+
52=20120403-20:43:01.837+55=FDAX+48=FDAX062012+167=FUT+207=Eurex+15=EUR+
320=12345+322=12345:1+200=201206+16451=0+393=4+323=4+16452=0.5+16899=0+
16454=25+16552=0.5+16554=25+16456=1+16457=1+16458=9999.99+146=0+18206=1+
18203=Eurex+864=1+865=5+866=20120614+10=051+
```

Response :

```
8=FIX.4.2+9=00285+35=d+49=TT_ORDER+56=REMOTE+34=26+
52=20120403-20:43:01.837+55=FDAX+48=FDAX032012+167=FUT+207=Eurex+15=EUR+
320=12345+322=12345:2+200=201203+16451=0+393=4+323=4+16452=0.5+16899=0+
16454=25+16552=0.5+16554=25+16456=1+16457=1+16458=9999.99+146=0+18206=1+
18203=Eurex+864=1+865=5+866=20120315+10=045+
```

Response :

```
8=FIX.4.2+9=00285+35=d+49=TT_ORDER+56=REMOTE+34=27+
52=20120403-20:43:01.837+55=FDAX+48=FDAX122012+167=FUT+207=Eurex+15=EUR+
320=12345+322=12345:3+200=201212+16451=0+393=4+323=4+16452=0.5+16899=0+
16454=25+16552=0.5+16554=25+16456=1+16457=1+16458=9999.99+146=0+18206=1+
18203=Eurex+864=1+865=5+866=20121220+10=043+
```

Thus, in this example, for prices less than 9999.99:

$$\begin{aligned} \text{Actual tick size} &= \text{Tag 16552 (ExchTickSize)} * \text{Tag 16457 (NumTicks)} \\ &= 0.5 * 1 \\ &= 0.5 \end{aligned}$$

Requesting information for energy securities

In this conversation, a FIX client requests energy securities.

Request :

```
8=FIX.4.2+9=96+35=c+34=2+49=REMOTE+52=20120425-21:06:59+56=TT_PRICE+
167=NRG+207=ICE_IPE+55=SP-15+320=1+17000=Y+10=165+
```

TT FIX Adapter responds by sending a **Security Definition (d)** message for each matching security. Because SP-15 futures can trade in flow, the response message body includes the flow information in Tags 16460 (DeliveryUnit), 16461 (LotSize), 16463 (Blocks), and 16464 (TradesInFlow).

Response :

```
8=FIX.4.2+9=00398+35=d+49=TT_PRICE+56=REMOTE+34=2+
52=20120425-21:06:59.739+55=SP-15+48=608735+10455=Cal 16+167=NRG+
207=ICE_IPE+15=USD+320=1+322=1:0+107=Fin Swap-Peak - SP15 DA+
16309=SPM SYF0016.Z0016+18207=SP-15+200=201512+205=30+16461=308+
16463=4928+16451=43+393=149+16464=Y+18211=Y+323=4+16460=25+16452=0.01+
16899=0+16454=616000+16552=0.05+16554=616000+16456=0+146=0+18206=1+
18203=ICE_IPE+864=1+865=5+866=20151230+10=213+
```

Response :

```
8=FIX.4.2+9=00396+35=d+49=TT_PRICE+56=REMOTE+34=3+52=20120425-
21:06:59.739+55=SP-15+48=604693+10455=Q3 12+167=NRG+207=ICE_IPE+15=USD+
320=1+322=1:1+107=Fin Swap-Peak - SP15 DA+16309=SPM SQN0012.U0012+
18207=SP-15+200=201206+205=28+16461=76+16463=1216+16451=43+393=149+
16464=Y+18211=Q+323=4+16460=25+16452=0.01+16899=0+16454=152000+
16552=0.05+16554=152000+16456=0+146=0+18206=1+18203=ICE_IPE+864=1+865=5+
866=20120628+10=248+
```

Response :

```
8=FIX.4.2+9=00396+35=d+49=TT_PRICE+56=REMOTE+34=4+
52=20120425-21:06:59.739+55=SP-15+48=604694+10455=Q4 12+167=NRG+
207=ICE_IPE+15=USD+320=1+322=1:2+107=Fin Swap-Peak - SP15 DA+
16309=SPM SQV0012.Z0012+18207=SP-15+200=201209+205=27+16461=77+
16463=1232+16451=43+393=149+16464=Y+18211=Q+323=4+16460=25+16452=0.01+
16899=0+16454=154000+16552=0.05+16554=154000+16456=0+146=0+18206=1+
18203=ICE_IPE+864=1+865=5+866=20120927+10=016+
```

Response :

```
8=FIX.4.2+9=00398+35=d+49=TT_PRICE+56=REMOTE+34=5+
52=20120425-21:06:59.739+55=SP-15+48=609024+10455=Cal 17+167=NRG+
207=ICE_IPE+15=USD+320=1+322=1:3+107=Fin Swap-Peak - SP15 DA+
16309=SPM SYF0017.Z0017+18207=SP-15+200=201612+205=29+16461=306+
16463=4896+16451=43+393=149+16464=Y+18211=Y+323=4+16460=25+16452=0.01+
16899=0+16454=612000+16552=0.05+16554=612000+16456=0+146=0+18206=1+
18203=ICE_IPE+864=1+865=5+866=20161229+10=226+
```


Requesting a security definition by symbol

The following conversation shows the response when a FIX client sends a **Security Definition Request (c)** message specifying only the symbol specified by the exchange.

Request :

```
8=FIX.4.2+9=88+35=c+34=3+49=Remote_p+52=20070212-19:58:31.802+
56=FA_price+55=IPE e-Brent+320=1+17000=Y+10=012+
```

TT FIX Adapter responds by sending a **Security Definition (d)** message for each matching security.

Response :

```
8=FIX.4.2+9=00239+35=d+49=FA_price+56=Remote_p+34=85+
52=20070212-19:58:31.818+55=IPE e-
Brent+48=216450+10455=Jun07+167=FUT+207=ICE_IPE+15=USD+320=1+322=1:0+200
=200705+16451=0+393=700+323=4+
16452=0.01+16899=200+16454=1000+16552=0.01+16554=1000+16456=0+146=0+
10=249+
```

Response :

```
8=FIX.4.2+9=00239+35=d+49=FA_price+56=Remote_p+34=86+
52=20070212-19:58:31.818+55=IPE e-
Brent+48=216646+10455=Dec07+167=FUT+207=ICE_IPE+15=USD+320=1+322=1:1+200
=200711+16451=0+393=700+323=4+
16452=0.01+16899=200+16454=1000+16552=0.01+16554=1000+16456=0+146=0+
10=222+
```

Response :

```
8=FIX.4.2+9=00239+35=d+49=FA_price+56=Remote_p+34=87+
52=20070212-19:58:31.818+55=IPE e-
Brent+48=216707+10455=Apr07+167=FUT+207=ICE_IPE+15=USD+320=1+322=1:2+200
=200703+16451=0+393=700+323=4+
16452=0.01+16899=200+16454=1000+16552=0.01+16554=1000+16456=0+146=0+
10=246+
```

Response :

```
8=FIX.4.2+9=00239+35=d+49=FA_price+56=Remote_p+34=88+
52=20070212-19:58:31.818+55=IPE e-
Brent+48=216709+10455=Aug07+167=FUT+207=ICE_IPE+15=USD+320=1+322=1:3+200
=200707+16451=0+393=700+323=4+
16452=0.01+16899=200+16454=1000+16552=0.01+16554=1000+16456=0+146=0+
10=248+
```

Requesting a security definition for options

The following conversation shows the response when a FIX client sends a **Security Definition Request (c)** message for options.

Request :

```
8=FIX.4.2+9=87+35=c+34=84+49=REMOTE+52=20120322-20:00:11+
56=TT_PRICE+167=OPT+55=OSA+320=12345+17000=Y+10=106+
```

TT FIX Adapter responds by sending a **Security Definition (d)** message for each matching security. This example shows only the first four responses.

Response :

```
8=FIX.4.2+9=00354+35=d+49=TT_PRICE+56=REMOTE+34=99+
52=20120322-20:00:11.089+55=OSA+48=00B0CN00OSA4059CZ+10455=OSAH3 P1030+
167=OPT+207=CBOT+15=USD+320=12345+322=12345:0+107=5 Year Interest Rate
Swap Options+200=201303+16451=16+393=502+201=0+323=4+202=103+
16452=0.015625+16899=0+16454=1000+16552=0.015625+16554=1000+
16456=0+146=0+18206=1+18203=CBOT+864=1+865=5+866=20130318+10=207+
```

Response :

```
8=FIX.4.2+9=00357+35=d+49=TT_PRICE+56=REMOTE+34=100+
52=20120322-20:00:11.089+55=OSA+48=00B0CN00OSA4059EZ+
10455=OSAH3 P1035+167=OPT+207=CBOT+15=USD+320=12345+322=12345:1+
107=5 Year Interest Rate Swap Options+200=201303+16451=16+393=502+
201=0+323=4+202=103.5+16452=0.015625+16899=0+16454=1000+16552=0.015625+
16554=1000+16456=0+146=0+18206=1+18203=CBOT+864=1+865=5+866=20130318+
10=092+
```

Response :

```
FIX.4.2+9=00357+35=d+49=TT_PRICE+56=REMOTE+34=101+52=20120322-
20:00:11.089+55=OSA+48=00B0CN00OSA405A2Z+10455=OSAH3 P1045+167=OPT+
207=CBOT+15=USD+320=12345+322=12345:2+107=5 Year Interest Rate Swap
Options+200=201303+16451=16+393=502+201=0+323=4+202=104.5+
16452=0.015625+16899=0+16454=1000+16552=0.015625+16554=1000+16456=0+
146=0+18206=1+18203=CBOT+864=1+865=5+866=20130318+10=085+
```

Response :

```
FIX.4.2+9=00357+35=d+49=TT_PRICE+56=REMOTE+34=101+
52=20120322-20:00:11.089+55=OSA+48=00B0CN00OSA405A2Z+10455=OSAH3 P1045+
167=OPT+207=CBOT+15=USD+320=12345+322=12345:2+107=5 Year Interest Rate
Swap Options+200=201303+16451=16+393=502+201=0+323=4+202=104.5+
16452=0.015625+16899=0+16454=1000+16552=0.015625+16554=1000+16456=0+
146=0+18206=1+18203=CBOT+864=1+865=5+866=20130318+10=085+
```

Sending an invalid exchange

The following conversation shows the response when a FIX client sends a **Security Definition Request (c)** message with an invalid exchange in Tag 207 (**SecurityExchange**). TT FIX Adapter drops the request and sends no response.

Request :

```
8=FIX.4.2+9=95+35=c+34=178+49=REMOTE+52=20120322-21:34:05+
56=TT_PRICE+167=FUT+55=ZB+207=XXX+320=12345+17000=Y+10=060
```

Response :

```
[no response]
```

TT FIX Adapter sends no response because a **Security Definition Request (c)** is actually a subscription. As the specified exchange might become available at some later time, TT FIX Adapter does not respond.

Security status

Overview

These examples show how TT FIX Adapter responds to a **Security Status Request (e)** message in the following scenarios:

- Requesting a security status snapshot
- Subscribing to status updates
- When you specify an invalid exchange
- Cancelling a status update subscription

Requesting a security status snapshot

In this conversation, a FIX client sends a **Security Status Request (e)** message to TT FIX Adapter requesting a status snapshot by setting Tag 263 (**SubscriptionRequestType**) to 0. It identifies the instrument by security ID, which uses Tags 48 (**SecurityID**), 55 (**Symbol**), and 207 (**SecurityExchange**).

Request:

```
8=FIX.4.2+9=95+35=e+34=4+49=REMOTE+52=20120323-21:30:19+
56=TT_PRICE+48=00A0IM00ZBZ+55=ZB+207=CBOT+263=1+324=3+10=067+
```

TT FIX Adapter responds with a **Security Status (f)** message with the current status of the security.

Response:

```
8=FIX.4.2+9=00137+35=f+49=TT_PRICE+56=REMOTE+34=3+
52=20120323-21:30:19.934+55=ZB+48=00A0IM00ZBZ+10455=ZBU2+167=FUT+
207=CBOT+15=USD+324=3+200=201209+326=18+10=143+
```

Subscribing to status updates

In this conversation, a FIX client sends a **Security Status Request (e)** message to TT FIX Adapter subscribing for status updates by setting Tag 263 (**SubscriptionRequestType**) to 1. It identifies the instrument by security name.

Request:

```
8=FIX.4.2+9=102+35=e+34=1256+49=REMOTE+52=20120323-15:22:22+
56=TT_PRICE+55=ZB+167=FUT+200=201206+207=CBOT+263=1+324=4+10=096+
```

TT FIX Adapter responds with a **Security Status (f)** message with the current status of the security. Additionally, as changes occur in the instrument, TT FIX Adapter sends subsequent **Security Status (f)** messages.

Response:

```
8=FIX.4.2+9=00140+35=f+49=TT_PRICE+56=REMOTE+34=1771+
52=20120323-15:22:23.012+55=ZB+48=00A0FM00ZBZ+10455=ZBM2+167=FUT+
207=CBOT+15=USD+324=4+200=201206+326=17+10=010+
```

When you specify an invalid exchange

The following conversation shows the response when a FIX client sends a **Security Status Request (e)** message with an unsupported exchange in Tag 207 (**SecurityExchange**).

Request :

```
8=FIX.4.2+9=95+35=e+34=4+49=REMOTE+52=20120323-21:30:19+
56=TT_PRICE+48=00A0IM00ZBZ+55=ZB+207=XXX+263=1+324=3+10=067+
```

TT FIX Adapter responds with a **Security Status (f)** message with Tag 326 (**SecurityTradingStatus**) set to 19, which indicates the exchange does not trade the product.

Response :

```
8=FIX.4.2+9=00099+35=f+49=TT_PRICE+56=REMOTE+34=2
+52=20120323-21:35:44.996+324=3+58=Unknown or missing market
+326=19+10=152+
```

Cancelling a status update subscription

In this conversation, a FIX client sends a **Security Status Request (e)** message to TT FIX Adapter to cancel status updates by setting Tag 263 (**SubscriptionRequestType**) to 2. TT FIX Adapter sends no response back to the FIX client.

Request :

```
8=FIX.4.2+9=95+35=e+34=4+49=REMOTE+52=20120323-21:30:19+
56=TT_PRICE+48=00A0IM00ZBZ+55=ZB+207=CBOT+263=21+324=3+10=067+
```

Response :

```
[No response sent]
```

Market data

Overview

These examples show how TT FIX Adapter responds to a **Market Data Request (v)** message in the following scenarios:

- [Subscribing for market data \(with no Bids or Offers\)](#)
- [Subscribing for market data \(with Bids and Offers\)](#)
- [Requesting incremental updates for an unsupported exchange](#)
- [Cancelling a market data subscription](#)
- [Cancelling a market data subscription with an unsupported exchange](#)
- [Requesting market data with market depth](#)

Subscribing for market data (with no Bids or Offers)

In this conversation, a FIX client sends TT FIX Adapter a **Market Data Request (v)** message to subscribe to Bids and Offers for the CME,FUT,6E,Jun12 and CBOT,FUT,ZT,SEP12 contracts by setting Tag 263 (**SubscriptionRequestType**) to 1 (snapshot and updates).

Request :

```
#8=FIX.4.2#9=166#35=V#34=2#49=REMOTE#52=20120409-
15:47:17#56=TT_PRICE#146=2#55=6E#48=00A0FM006EZ#207=CME#55=ZT#48=00A0IM
00ZTZ#207=CBOT#262=2#263=1#264=1#265=1#266=Y#267=2#269=0#269=1#10=042#
```

Initially, TT FIX Adapter responds by sending two **Market Data Snapshot Full Refresh (w)** messages, one for each instrument. The first response shows Tag 268 (**NoMDEntries**) as 0, indicating that no market data is available for the 6E Jun12 contract. The second response shows that no market data is available for the ZT SEP12 contract. The response also includes Tag 387 (**TotalVolumeTraded**), if volume exists.

Response :

```
200=200306#207=TTSIM#262=2#268=0#10455=Jun03#10=072#8=FIX.4.2#9=00150#3
5=W#49=TT_PRICE#56=REMOTE#34=2#52=20120409-
15:47:17.195#55=6E#48=00A0FM006EZ#10455=6EM2#167=FUT#207=CME#15=USD#262
=2#200=201206#18210=1#387=30#268=0#10=061#
```

Response :

```
8=FIX.4.2#9=00144#35=W#49=TT_PRICE#56=REMOTE#34=3#52=20120409-
15:47:17.195#55=ZT#48=00A0IM00ZTZ#10455=ZTU2#167=FUT#207=CBOT#15=USD#26
2=2#200=201209#18210=1#268=0#10=248#
```

As the market moves for the contracts, TT FIX Adapter sends **Market Data Incremental Refresh (x)** messages as changes occur.

Response :

```
8=FIX.4.2#9=00175#35=X#49=TT_PRICE#56=REMOTE#34=18#52=20120409-
16:01:35.705#262=2#268=1#279=0#55=6E#48=00A0FM006EZ#10455=6EM2#167=FUT#
207=CME#15=USD#200=201206#18210=1#269=0#270=6#271=1#387=30#10=166#
```

Response :

```
8=FIX.4.2+9=00205+35=X+49=TT_PRICE+56=REMOTE+34=19+52=20120409-
16:01:45.002+262=2+268=2+279=2+55=6E+48=00A0FM006EZ+10455=6EM2+167=FUT+
207=CME+15=USD+200=201206+290=1+18210=1+269=0+270=6+271=1+387=30+279=0+
269=0+270=5+271=1+10=217+
```

Response :

```
8=FIX.4.2+9=00178+35=X+49=TT_PRICE+56=REMOTE+34=35+52=20120409-
16:14:06.734+262=2+268=1+279=0+55=ZT+48=00A0IM00ZTZ+10455=ZTU2+167=FUT+
207=CBOT+15=USD+200=201209+18210=1+269=0+270=109.953125+271=5+10=050+
```

Subscribing for market data (with Bids and Offers)

In this conversation, a FIX client sends TT FIX Adapter a **Market Data Request (v)** message to subscribe to Bids and Offers for the CME,FUT,ES,SEP12 and CBOT,FUT,ZB,SEP12 contracts by setting Tag 263 (**SubscriptionRequestType**) to 1 (snapshot and updates).

Request :

```
8=FIX.4.2+9=166+35=V+34=5+49=REMOTE+52=20120326-15:50:16+
56=TT_PRICE+146=2+55=ES+48=00A0IM00ESZ+207=CME+55=ZB+48=00A0IM00ZBZ+
207=CBOT+262=2+263=1+264=1+265=1+266=Y+267=2+269=0+269=1+10=061+
```

Initially, TT FIX Adapter responds by sending two **Market Data Snapshot Full Refresh (w)** messages, one for each instrument, showing the following:

- For the ES,SEP12 contract, a bid for 1 at 139500 and an offer for 1 at 139475
- For the ZB,SEP12 contract, an offer for 2 at 136.84375

The response includes Tag 387 (**TotalVolumeTraded**) to show the volume, if it exists.

Response :

```
8=FIX.4.2+9=00215+35=W+49=TT_PRICE+56=REMOTE+34=5+
52=20120326-15:50:16.652+55=ES+48=00A0IM00ESZ+10455=ESU2+167=FUT+
207=CME+15=USD+262=2+200=201209+18210=1+387=8448+268=2+269=0+290=1+
270=139475+271=1738+269=1+290=1+270=139500+271=450+10=127+
```

Response :

```
8=FIX.4.2+9=00183+35=W+49=TT_PRICE+56=REMOTE+34=6+
52=20120326-15:50:16.652+55=ZB+48=00A0IM00ZBZ+10455=ZBU2+167=FUT+
207=CBOT+15=USD+262=2+200=201209+18210=1+387=17+268=1+269=1+290=2+
270=136.84375+271=5+10=215+
```

As the market moves for the contracts, TT FIX Adapter sends **Market Data Incremental Refresh (x)** messages as changes occur. In this conversation, both contracts show activity.

Response :

```
8=FIX.4.2+9=00189+35=X+49=TT_PRICE+56=REMOTE+34=7+
52=20120326-15:50:29.292+262=2+268=1+279=1+55=ES+48=00A0IM00ESZ+
10455=ESU2+167=FUT+207=CME+15=USD+200=201209+290=1+18210=1+269=1+
270=139500+271=400+387=8448+10=195+
```

Response :

```
8=FIX.4.2+9=00190+35=X+49=TT_PRICE+56=REMOTE+34=8+
2=20120326-15:51:03.949+262=2+268=1+279=1+55=ES+48=00A0IM00ESZ+
10455=ESU2+167=FUT+207=CME+15=USD+200=201209+290=1+18210=1+269=0+
270=139475+271=1705+387=8481+10=254+
```

Response :

```
8=FIX.4.2+9=00190+35=X+49=TT_PRICE+56=REMOTE+34=34+
52=20120326-16:01:09.792+262=2+268=1+279=2+55=ZB+48=00A0IM00ZBZ+
10455=ZBU2+167=FUT+207=CBOT+15=USD+200=201209+290=1+18210=1+269=1+
270=136.84375+271=5+387=22+10=025+
```

Requesting incremental updates for an unsupported exchange

In this conversation, a FIX client sends TT FIX Adapter a **Market Data Request (v)** message for market data for an unsupported exchange (**XXX**). TT FIX Adapter responds by sending a **Market Data Request Reject (y)** message with the reject reason in Tag 58 (**Text**).

Request :

```
8=FIX.4.2+9=137+35=V+34=44+49=REMOTE+52=20120326-16:28:17+56=TT_PRICE+
146=1+48=00A0IM00ZBZ+55=ZB+207=XXX+262=2+263=0+264=1+265=1+266=Y+267=2+
269=0+269=1+10=004+
```

Response :

```
8=FIX.4.2+9=00103+35=Y+49=TT_PRICE+56=REMOTE+34=77+
52=20120326-16:28:17.042+262=2+58=Unknown or missing market: Entry
#1+10=077+
```

Cancelling a market data subscription

In this conversation, a FIX client sends TT FIX Adapter a **Market Data Request (v)** message. By setting Tag 263 (**SubscriptionRequestType**) to 2 (unsubscribe), the FIX client asks TT FIX Adapter to cancel market data updates.

Request :

```
8=FIX.4.2+9=166+35=V+34=5+49=REMOTE+52=20120326-15:50:16+
56=TT_PRICE+146=2+55=ES+48=00A0IM00ESZ+207=CME+55=ZB+
48=00A0IM00ZBZ+207=CBOT+262=2+263=2+264=1+265=1+266=Y+267=2+269=0+
269=1+10=061+
```

Response :

[no response]

Canceling a market data subscription with an unsupported exchange

In this conversation, a FIX client sends TT FIX Adapter a **Market Data Request (v)** message for an unsupported exchange (**XXX**). By setting Tag 263 (**SubscriptionRequestType**) to 2, the FIX client asks TT FIX Adapter to cancel market data updates. TT FIX Adapter responds by sending a **Market Data Request Reject (x)** message with the reject reason in Tag 58 (**Text**)

Request :

```
8=FIX.4.2+9=131+35=V+34=62+49=REMOTE+52=20120326-16:41:51+56=TT_PRICE+
146=1+48=00A0IM00ZBZ+55=ZB+207=XXX+262=4+263=2+264=1+265=1+266=Y+267=2+
269=0+10=235+
```

Response :

```
8=FIX.4.2+9=00103+35=Y+49=TT_PRICE+56=REMOTE+34=95+
52=20120326-16:41:51.183+262=4+58=Unknown or missing market: Entry
#1+10=078+
```

Requesting market data with market depth

In this conversation, a FIX client sends TT FIX Adapter a **Market Data Request (v)** message to subscribe to Bid and Ask for the CME,ES,FUT,SEP12 contract. By setting Tag 264 (**MarketDepth**) to 5, the FIX client asks TT FIX Adapter to send five levels of depth.

Request :

```
8=FIX.4.2+9=137+35=V+34=67+49=REMOTE+52=20120326-16:46:16+56=TT_PRICE+
146=1+55=ES+48=00A0IM00ESZ+207=CME+262=8+263=1+264=5+265=1+266=Y+267=2+
269=0+269=1+10=216+
```

Initially, TT FIX Adapter responds by sending a **Market Data Snapshot Full Refresh (w)** message with the current volume traded in Tag 387 (**TotalVolumeTraded**). This example contains a bid for 562 at 139475 and an offer for 25 at 139500.

Response :

```
8=FIX.4.2+9=00215+35=W+49=TT_PRICE+56=REMOTE+34=100+
52=20120326-16:46:16.605+55=ES+48=00A0IM00ESZ+10455=ESU2+167=FUT+
207=CME+15=USD+262=8+200=201209+18210=1+387=9800+268=2+269=0+290=1+
270=139475+271=562+269=1+290=1+270=139500+271=25+10=118+
```

As the market moves for the contract, TT FIX Adapter sends **Market Data Incremental Refresh (x)** messages as changes occur.

Response :

```
8=FIX.4.2+9=00190+35=X+49=TT_PRICE+56=REMOTE+34=102+
52=20120326-16:46:16.730+262=8+268=1+279=1+55=ES+48=00A0IM00ESZ+
10455=ESU2+167=FUT+207=CME+15=USD+200=201209+290=1+18210=1+269=1+
270=139500+271=26+387=9800+10=233+
```


Response :

```
8=FIX.4.2+9=00190+35=X+49=TT_PRICE+56=REMOTE+34=103+
52=20120326-16:46:16.730+262=8+268=1+279=1+55=ES+48=00A0IM00ESZ+
10455=ESU2+167=FUT+207=CME+15=USD+200=201209+290=1+18210=1+269=1+
270=139500+271=27+387=9800+10=235+
```

New order

Overview

These examples show how TT FIX Adapter responds to a **New Order Single (D)** message in the following scenarios:

- [Submitting a new order and requesting summary fills](#)
- [Submitting a new order with multiple flavors of a TT Gateway](#)
- [Submitting a new order with Tag 18205 \(TTAccountType\)](#)
- [Submitting an order with CME mandatory iLink tags](#)
- [Submitting a calendar spread order by security name](#)
- [Submitting an order when configured to send security legs](#)
- [Submitting an incomplete order](#)
- [Submitting an invalid order with restricted account defaults](#)
- [Submitting an order that exceeds the credit limit](#)
- [Submitting an order that exceeds the position limits](#)
- [Submitting an order that exceeds the maximum quantity](#)

Submitting a new order and requesting summary fills

This conversation shows how TT FIX Adapter responds to new orders when the TT FIX Adapter configuration enables the **Send Summary Fills** setting for the FIX session. In this conversation, a FIX client sends a **New Order Single (D)** message to TT FIX Adapter to submit an order to sell 12 Eurex,MLEG,FDAX,JUN12/SEP12 contracts at 3.5.

Request:

```
8=FIX.4.2+9=157+35=D+34=2+49=REMOTE+52=20120329-20:36:44+56=TT_ORDER+
1=ps001001+11=443184512+38=12+40=2+44=3.5+47=A+48=FDAX062012SPD092012+5
4=2+77=O+204=1+207=Eurex+55=FDAX+10=030+
```

TT FIX Adapter responds with an **Execution Report (8)** message confirming the order. In this example, the TT FIX Adapter configuration enables the **Send Security Legs** setting for the FIX session, so it includes the information about the security and its underlying instruments.

Response:

```
8=FIX.4.2+9=00527+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+
57=NONE+34=2+52=20120329-20:36:44.536+55=FDAX+48=FDAX062012SPD092012+
10762=Calendar+167=MLEG+207=Eurex+15=EUR+1=ps001001+47=A+
204=1+10553=REMOTE+11=443184512+18203=Eurex+18216=6874+37=022ZL2001+
17=022ZL2001:0+198=0009MQ469+151=12+14=0+54=2+40=2+77=O+59=0+11028=N+
150=0+20=0+39=0+442=3+44=3.5+38=12+6=0+60=20120329-20:36:44.380+146=2+
311=FDAX+309=FDAX062012+310=FUT+308=Eurex+318=EUR+313=201206+319=1+
16624=1+311=FDAX+309=FDAX092012+310=FUT+308=Eurex+318=EUR+313=201209+31
9=1+16624=2+10=181+
```

After the order gets filled, TT FIX Adapter responds with two **Execution Report (8)** messages confirming the leg fills. Later it sends another **Execution Report (8)** message with summary fill information.

Response :

```
8=FIX.4.2+9=00458+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=6+52=20120329-20:38:11.067+55=FDAX+48=FDAX062012+167=FUT+207=Eurex+
15=EUR+1=ps001001+47=A+204=1+10553=REMOTE+11=454446059+18203=Eurex+
18216=6874+37=022XKT037+17=L00003R000000009MQ46829S+58=Leg Fill+
198=0009MQ468+10527=00003R+16018=m1m7s0+200=201206+32=12+151=0+14=12+
54=2+40=2+77=0+59=0+11028=N+150=2+20=0+39=2+442=2+44=3.5+38=12+31=7019+
6=7019+60=20120329-20:38:10.970+6038=20120329-20:33:45.460+146=0+
10=117+
```

Response :

```
8=FIX.4.2+9=00462+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=7+52=20120329-20:38:11.067+55=FDAX+48=FDAX092012+167=FUT+207=Eurex+
15=EUR+1=ps001001+47=A+204=1+10553=REMOTE+11=454446059+18203=Eurex+
18216=6874+37=022XKT037+17=L00003S000000009MQ46829B+58=Leg Fill+
198=0009MQ468+10527=00003S+16018=m1m7s0+200=201209+32=12+151=0+14=12+
54=1+40=2+77=0+59=0+11028=N+150=2+20=0+39=2+442=2+44=3.5+38=12+
31=7015.5+6=7015.5+60=20120329-20:38:10.970+6038=20120329-20:33:45.460+
146=0+10=037+
```

Response :

```
8=FIX.4.2+9=00624+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+
57=NONE+34=5+52=20120329-20:38:11.067+55=FDAX+48=FDAX062012SPD092012+
10762=Calendar+167=MLEG+207=Eurex+15=EUR+1=ps001001+47=A+204=1+
10553=REMOTE+11=454446059+18203=Eurex+18216=6874+37=022XKT037+
17=S00003R000000009MQ46829S+58=Summary Fill+198=0009MQ468+10527=00003R+
16018=m1m7s0+32=12+151=0+14=12+54=2+40=2+77=0+59=0+11028=N+150=2+
20=0+39=2+442=3+44=3.5+38=12+31=3.5+6=3.5
+60=20120329-20:38:10.970+6038=20120329-20:33:45.460+146=2+311=FDAX+
309=FDAX062012+310=FUT+308=Eurex+318=EUR+313=201206+319=1+16624=1+
311=FDAX+309=FDAX092012+310=FUT+308=Eurex+318=EUR+313=201209+319=1+
16624=2+10=146+
```

Submitting a new order with multiple flavors of a TT Gateway

This conversation shows how to route new orders when a FIX client user is configured to connect to two (or more) flavors of the same TT Gateway, and the contract for which the order is being routed is available on more than one of these TT Gateways.

FIX clients can determine the list of TT Gateways on which a contract is available by performing a **Security Definition Request (c)**. The resultant **Security Definition (d)** messages contain this information in Tag 18206 (**NoGateways**) and Tag 18203 (**ExchangeGateway**).

Request :

```
8=FIX.4.2+9=81+35=c+34=2+49=REMOTE+52=20120426-15:30:12+56=TT_ORDER+
167=FUT+207=CME+55=ES+320=1+10=064+
```

TT FIX Adapter responds by sending a **Security Definition (d)** message for each matching security. The following example shows only the ES Jun12 FUT security.

Response :

```
8=FIX.4.2+9=00272+35=d+49=TT_ORDER+56=REMOTE+34=3+
52=20120426-15:30:12.603+55=ES+48=00A0FM00ESZ+10455=ESM2+167=FUT+
207=CME+15=USD+320=1+322=1:1+200=201206+16451=43+393=5+323=4+16452=5+
16899=0+16454=0.5+16552=25+16554=0.5+16456=0+146=0+18206=2+
18203=CME-B+18203=CME+864=1+865=5+866=20120615+10=109+
```

Assume we want to route an order to the ES Jun12 FUT contract, which exists on both the CME and the CME-B TT Gateway flavors. If Tag 18203 (**ExchangeGateway**) is not provided, the order will be rejected.

Request :

```
8=FIX.4.2+9=168+35=D+34=2+49=REMOTE+52=20120426-20:04:49+56=TT_ORDER+
1=ps001001+11=491304618+38=10+40=2+44=136000+47=A+54=1+55=ES+
60=20051205-09:11:59+200=201206+167=FUT+204=0+207=CME+10=190+
```

TT FIX Adapter sends an **Execution Report (8)** message with Tag 103 (**OrdRejReason**) showing the rejection code and Tag 58 (**Text**) showing the reason.

Response :

```
8=FIX.4.2+9=00334+35=8+49=TT_ORDER+56=REMOTE+50=NONE+57=NONE+34=2+
52=20120426-20:04:49.738+55=ES+167=FUT+207=CME+1=ps001001+47=A+204=0+
11=491304618+37=NONE+17=0SD88X002+58=Order routable on multiple
gateways. Exchange gateway must be provided+200=201206+32=0+103=0+151=0+
14=0+54=1+40=2+150=8+20=0+39=8+44=136000+38=10+31=0+6=0+
60=20120426-20:04:49.738+10=217+
```

Therefore, The **New Order Single (D)** message must provide Tag 18203 (**ExchangeGateway**) to indicate whether the order should be routed to the CME or CME-B TT Gateway flavor.

Request :

```
8=FIX.4.2+9=180+35=D+34=4+49=REMOTE+52=20120426-20:05:58+56=TT_ORDER+
1=ps001001+11=584303007+38=10+40=2+44=136000+47=A+54=1+55=ES+
60=20051205-09:11:59+200=201206+167=FUT+204=0+207=CME+
18203=CME-B+10=053+
```

TT FIX Adapter responds with an **Execution Report (8)** message confirming the order.

Response :

```
8=FIX.4.2+9=00368+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=4+52=20120426-20:05:59.202+55=ES+48=00A0FM00ESZ+10455=ESM2+167=FUT+
207=CME+15=USD+1=ps001001+47=A+204=0+10553=REMOTE+11=584303007+
18203=CME-B+16142=US,IL+37=0SD88X003+17=0SD88X003:0+198=00000B9E+
200=201206+151=10+14=0+54=1+40=2+77=0+59=0+11028=N+150=0+20=0+39=0+
442=1+44=136000+38=10+6=0+60=20120426-20:05:58.615+10=042+
```

Submitting a new order with Tag 18205 (TTAccountType)

This conversation shows how to route new orders when the **Send Account Type As** configuration option in FACT is set to *Tag 18205*. In such a scenario, Tag 18205 (**TTAccountType**) is required, while Tag 47 (**Rule80A**) and Tag 204 (**CustomerOrFirm**) are not sent.

Request:

```
8=FIX.4.2+9=166+35=D+34=4+49=REMOTE+52=20120426-19:47:31+56=TT_ORDER+
1=ps001001+11=311899333+38=10+40=2+44=136000+54=1+55=ES+
60=20051205-09:11:59+200=201206+167=FUT+18205=A1+207=CME+10=138+
```

TT FIX Adapter responds with an **Execution Report (8)** message confirming the order.

Response:

```
8=FIX.4.2+9=00373+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=4+52=20120426-19:47:31.812+55=ES+48=00A0FM00ESZ+10455=ESM2+167=FUT+
207=CME+15=USD+1=ps001001+18205=A1+10553=REMOTE+11=311899333+18203=CME+
16142=US,IL+18216=P15000+37=023DNX001+17=023DNX001:0+198=SCZQ+
200=201206+151=10+14=0+54=1+40=2+77=O+59=0+11028=N+150=0+20=0+39=0+
442=1+44=136000+38=10+6=0+60=20120426-19:47:31.306+10=131+
```

Submitting an order with CME mandatory iLink tags

This conversation shows how the FIX client application can specify Tag 11028 (**ManualOrderIndicator**) and/or Tag 16142 (**OrderOriginationID**) per order to override the default values for the CME Mandatory iLink Tags. By default, orders submitted by TT FIX Adapter will be considered automated and the trader's location will be populated based on the FIX client user's location information from TT User Setup. Note that TT FIX Adapter will send Tag 11028 (**ManualOrderIndicator**) to any TT Gateway, while Tag 16142 (**OrderOriginationID**) will only be sent to TT CME and TT CBOT Gateways. If the FIX client wanted to route an order to a CME Gateway and specify both the trader's location and that it is a manual order, the FIX client will send Tag 11028 (**ManualOrderIndicator**) = *Y* and Tag 16142 (**OrderOriginationID**) = *US,IL* (for this example).

Request:

```
8=FIX.4.2+9=188+35=D+34=4+49=REMOTE+52=20120426-19:39:47+56=TT_ORDER+
1=ps001001+11=474119375+38=10+40=2+44=136000+47=A+54=1+55=ES+60=2005120
5-09:11:59+200=201206+167=FUT+204=0+207=CME+11028=Y+16142=US,IL+10=013+
```

TT FIX Adapter responds with an **Execution Report (8)** message confirming the order.

Response:

```
8=FIX.4.2+9=00375+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=5+52=20120426-19:39:48.008+55=ES+48=00A0FM00ESZ+10455=ESM2+
167=FUT+207=CME+15=USD+1=ps001001+47=A+204=0+10553=REMOTE+11=474119375+
18203=CME+16142=US,IL+18216=P15000+37=023DNJ002+17=023DNJ002:0+
198=SCZP+200=201206+151=10+14=0+54=1+40=2+77=O+59=0+11028=Y+150=0+20=0+
39=0+442=1+44=136000+38=10+6=0+60=20120426-19:39:47.510+10=192+
```

Submitting a calendar spread order by security name

In this conversation, the FIX client sends a calendar spread order. Instead of identifying the spread by its exchange security ID in Tag 48 (**SecurityID**), this message identifies the calendar spread by its security name and its underlying instruments.

Request:

```
8=FIX.4.2+9=254+35=D+56=TT_ORDER+49=REMOTE+34=5366+
52=20120403-14:51:19+11=192388462+47=A+204=0+1=ps001001+55=ES+
167=MLEG+207=CME+10762=Calendar+146=2+311=ES+308=CME+310=FUT+
313=201206+16624=2+319=1+311=ES+308=CME+310=FUT+313=201209+319=1+
16624=1+40=2+44=-590+38=1+54=1+10=022+
```

TT FIX Adapter responds with an **Execution Report (8)** message confirming the order. In this example, the TT FIX Adapter configuration disables the **Send Security Legs** setting for the FIX session, so it includes the exchange ID for the calendar spread in Tag 48 (**SecurityID**) instead of all the leg information.

Response:

```
8=FIX.4.2+9=00564+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+
57=NONE+34=5366+52=20120403-14:51:19.769+55=ES+
48=00CES200A0FM00ESZ1100A0IM00ESZ1+10455=ESM2-ESU2+
10762=Calendar+167=MLEG+207=CME+15=USD+1=ps001001+47=A+204=0+
10553=REMOTE+11=192388462+18203=CME+18216=P15000+37=02305J012+
17=02305J012:0+198=R0U3+151=1+14=0+54=1+40=2+77=0+59=0+11028=N+
150=0+20=0+39=0+442=3+44=-590+38=1+6=0+60=20120403-14:51:19.351+
146=2+311=ES+309=00A0FM00ESZ+310=FUT+308=CME+10456=ESM2+318=USD+
313=201206+319=1+16624=2+311=ES+309=00A0IM00ESZ+310=FUT+308=CME+
10456=ESU2+318=USD+313=201209+319=1+16624=1+10=012+
```

Submitting an order when configured to send security legs

This conversation shows how TT FIX Adapter responds to new orders when the TT FIX Adapter configuration enables the **Send Summary Fills** setting for the FIX session. In this conversation, a FIX client sends a **New Order Single (d)** message to TT FIX Adapter to submit an order to buy 12 Eurex,MLEG,FDAX,JUN12/SEP12 contracts at 3.5.

Request:

```
8=FIX.4.2+9=157+35=D+34=5+49=REMOTE+52=20120329-20:51:58+56=TT_ORDER+
1=ps001001+11=584533403+38=12+40=2+44=3.5+47=A+48=FDAX062012SPD092012+
54=1+77=0+204=1+207=Eurex+55=FDAX+10=037+
```

TT FIX Adapter responds with a series of **Execution Report (8)** messages containing the underlying leg information. Tag 146 (**NoRelatedSym**) indicates the contract contains two legs.

Response:

```
8=FIX.4.2+9=00528+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=11+52=20120329-20:51:58.640+55=FDAX+48=FDAX062012SPD092012+
10762=Calendar+167=MLEG+207=Eurex+15=EUR+1=ps001001+47=A+204=1+
10553=REMOTE+11=584533403+18203=Eurex+18216=6874+37=022ZL2006+
17=022ZL2006:0+198=0009MQ46K+151=12+14=0+54=1+40=2+77=0+59=0+11028=N+
150=0+20=0+39=0+442=3+44=3.5+38=12+6=0+60=20120329-20:51:58.500+
146=2+311=FDAX+309=FDAX062012+310=FUT+308=Eurex+318=EUR+313=201206+
319=1+16624=1+311=FDAX+309=FDAX092012+310=FUT+308=Eurex+318=EUR+
313=201209+319=1+16624=2+10=254+
```

Submitting an incomplete order

In this conversation, a FIX client sends a **New Order Single (D)** message to TT FIX Adapter to submit an order to buy 12 Eurex,MLEG,FDAX,Jun12/Sep12 contracts, but omits required Tag 44 (**Price**).

Request :

```
8=FIX.4.2#9=150#35=D#34=7#49=REMOTE#52=20120329-
20:53:42#56=TT_ORDER#1=ps001001#11=429368883#38=12#40=2#47=A#48=FDAX062
012SPD092012#54=1#77=O#204=1#207=Eurex#55=FDAX#10=239#
```

TT FIX Adapter sends an **Execution Report (8)** message with Tag 103 (**OrdRejReason**) showing the rejection code and Tag 58 (**Text**) showing the reason.

Response :

```
8=FIX.4.2#9=00285#35=8#49=TT_ORDER#56=REMOTE#50=NONE#57=NONE#34=13#
52=20120329-20:53:42.936#55=FDAX#48=FDAX062012SPD092012#207=Eurex#
1=ps001001#47=A#204=1#11=429368883#37=NONE#17=022ZL2007#
58=Missing Price (44)#32=0#103=0#151=0#14=0#54=1#40=2#77=O#150=8#20=0#
39=8#38=12#31=0#6=0#60=20120329-20:53:42.936#10=129#
```

Submitting an invalid order with restricted account defaults

In this conversation, a FIX client sends a **New Order Single (D)** message to buy 1 CME,MLEG,ES,Jun12/Sep12 spread contract at -590. In TT User Setup, the TT FIX Adapter client is configured with locked account defaults, and an **Account Type** of U3, but the order sets Tag 47 (**Rule80A**) to A.

Request :

```
8=FIX.4.2#9=254#35=D#56=TT_ORDER#49=REMOTE#34=5372#
52=20120403-14:56:57#11=576427210#47=A#204=0#1=ps001001#55=ES#167=MLEG#
207=CME#10762=Calendar#146=2#311=ES#308=CME#310=FUT#313=201206#16624=2#
319=1#311=ES#308=CME#310=FUT#313=201209#319=1#16624=1#40=2#44=-590#
38=1#54=1#10=017#
```

TT FIX Adapter sends an **Execution Report (8)** message with Tag 150 (**ExecType**) set to 8 (rejected) and Tag 58 (**Text**) showing the reason.

Response :

```
8=FIX.4.2#9=00329#35=8#49=TT_ORDER#56=REMOTE#50=NONE#57=NONE#
34=5372#52=20120403-14:56:57.658#55=ES#10762=Calendar#167=MLEG#
207=CME#1=ps001001#47=A#204=0#11=576427210#37=NONE#17=02305J013#
58=Account Defaults locked - AccountType does not match default#32=0#
103=0#151=0#14=0#54=1#40=2#150=8#20=0#39=8#44=-590#38=1#31=0#6=0#
60=20120403-14:56:57.658#10=232#
```

Submitting an order that exceeds the credit limit

This conversation shows the FIX messages exchanged when a trader tries to execute a trade that exceeds the credit limit. This example assumes that Guardian uses client-side risk checking and that the trader is near the established credit limit.

The FIX client sends a **New Order Single (D)** message to TT FIX Adapter to submit an order to buy one contract.

Request :

```
8=FIX.4.2+9=149+35=D+34=1112+49=REMOTE+52=20120405-13:56:47+
56=TT_ORDER+1=ps001001+11=476625378+38=1+40=2+44=13050+47=A+54=1+55=6E+
167=FUT+200=201209+204=0+207=CME+10=008+
```

TT FIX Adapter sends an **Execution Report (8)** message with Tag 150 (**ExecType**) set to 8 (rejected) and Tag 58 (**Text**) showing the reason. The "From Gateway:" at the beginning of the tag indicates the message originates from the gateway and not TT FIX Adapter. For more information about this text message, see Tag 58 (Text), on page 66.

Response :

```
8=FIX.4.2+9=00467+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=1138+52=20120405-13:56:47.866+55=6E+48=00A0IM006EZ+10455=6EU2+
167=FUT+207=CME+15=USD+1=ps001001+47=A+204=0+10553=REMOTE+11=476625378+
18203=CME+37=0300G7016+17=0300G7016:0+58=From Gateway: REMOTE Excds
CrLmt (LMT100+PL-64983.01-M1320.00=AV-66203.01)USD P1 B10S0F0NP10LP10SP0
X0912 B10S0F0WP10+200=201209+103=0+151=0+14=0+54=1+40=2+77=0+59=0+150=8+
20=0+39=8+442=1+44=13050+38=10+6=0+60=20120405-13:56:47.866+146=0+
10=017+
```

Submitting an order that exceeds the position limits

This conversation shows the FIX messages exchanged when a trader tries to execute a trade that exceed the credit limit. This example assumes that Guardian uses client-side risk checking and that the trader is near the established position limit.

The FIX client sends a **New Order Single (D)** message to TT FIX Adapter to submit an order to by 1 Eurex, FUT ,FDAX, Sep12 contract at 10.

Request :

```
8=FIX.4.2+9=152+35=D+34=15+49=REMOTE+52=20120404-19:12:33+
56=TT_ORDER+1=ps001001+11=339162646+38=10+40=2+44=7075.5+47=A+54=1+
55=FDAX+167=FUT+200=201209+204=0+207=Eurex+10=172+
```

TT FIX Adapter sends an **Execution Report (8)** message with Tag 150 (**ExecType**) set to 8 (rejected) and Tag 58 (**Text**) showing the reason. The "From Gateway:" at the beginning of the tag indicates the message originates from the gateway and not TT FIX Adapter. For more information about this text message, see Tag 58 (Text), on page 66.

Response :

```
8=FIX.4.2+9=00426+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=15+52=20120404-19:12:33.963+55=FDAX+48=FDAX092012+167=FUT+
207=Eurex+15=EUR+1=ps001001+47=A+204=0+10553=REMOTE+
11=339162646+18203=Eurex+37=OSD2QT012+17=OSD2QT012:0+58=From Gateway:
WCNPP(105) > MxNPP(100) P1 B110S0F-5NP105LP110SP5 X0912 B10S0F0WP100+
200=201209+103=0+151=0+14=0+54=1+40=2+77=0+59=0+150=8+20=0+39=8+442=1+
44=7075.5+38=10+6=0+60=20120404-19:12:33.963+146=0+10=254+
```


Submitting an order that exceeds the maximum quantity

This conversation shows the FIX messages exchanged when a trader tries to execute an order with a quantity larger than his maximum allowed order quantity. This example assumes that Guardian uses client-side risk checking and that maximum order quantity for Eurex FDAX contracts is 10.

The FIX client sends a **New Order Single** (D) message to TT FIX Adapter to submit an order to buy 25 Eurex, FUT, FDAX, Sep12 contracts.

Request :

```
8=FIX.4.2+9=147+35=D+34=39+49=REMOTE+52=20120404-20:05:46+56=TT_ORDER+
1=ps001001+11=4657571+38=25+40=2+44=138+47=A+54=1+55=FDAX+167=FUT+
200=201206+204=0+207=Eurex+10=184+
```

TT FIX Adapter sends an **Execution Report** (8) message with Tag 150 (**ExecType**) set to 8 (rejected) and Tag 58 (**Text**) showing the reason. The "From Gateway:" at the beginning of the tag indicates the message originates from the gateway and not TT FIX Adapter. For more information about this text message, see Tag 58 (Text), on page 66.

Response :

```
8=FIX.4.2+9=00416+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=61+52=20120404-20:05:46.115+55=FDAX+48=FDAX062012+
167=FUT+207=Eurex+15=EUR+1=ps001001+47=A+204=0+10553=REMOTE+11=4657571+
18203=Eurex+37=0300G7013+17=0300G7013:0+58=From Gateway: ORDQ25 >
OQLMT10 P1 B25S42F-13NP55LP12SP55 X0612 B25S42F-13WP-55+
200=201206+103=0+151=0+14=0+54=1+40=2+77=0+59=0+150=8+20=0+39=8+442=1+
44=138+38=25+6=0+60=20120404-20:05:46.115+146=0+10=056+
```

Change order

Overview

These examples show how TT FIX Adapter responds to **Order Cancel Request (F)** and **Order Cancel Replace Request (G)** messages in the following scenarios:

- [Changing the price of an order](#)
- [Canceling an order](#)
- [Canceling a filled order](#)
- [Changing a filled order](#)

Changing the price of an order

In this conversation, a FIX client sends a **New Order Single (D)** message to TT FIX Adapter to submit an order to buy 100 CME,FUT,ES,Jun12 contracts at 130000. TT FIX Adapter responds with an **Execution Report (8)** message confirming the order.

Request:

```
8=FIX.4.2#9=168#34=5#49=REMOTE#52=20120327-16:39:50#
56=TT_ORDER#1=ps001001#11=50351715#38=100#40=2#44=130000#47=A#54=1#
55=ES#60=20051205-09:11:59#200=201206#167=FUT#204=0#207=CME#10=183#
```

Response:

```
8=FIX.4.2#9=00364#35=8#49=TT_ORDER#56=REMOTE#50=TTORDPS001001#57=NONE#
34=5#52=20120327-16:39:50.457#55=ES#48=00A0FM00ESZ#10455=ESM2#167=FUT#
207=CME#15=USD#1=ps001001#47=A#204=0#10553=REMOTE#11=50351715#
18203=CME#18216=P15000#37=022XBJ002#17=022XBJ002:0#198=QZ1P#200=201206#
151=100#14=0#54=1#40=2#77=O#59=0#11028=N#150=0#20=0#39=0#442=1#
44=130000#38=100#6=0#60=20120327-16:39:50.423#10=028#
```

Before any of the order gets filled, the FIX client sends a **Order Cancel Replace Request (G)** message to change the price to 140625.

Request:

```
8=FIX.4.2#9=171#35=G#34=20#49=REMOTE#52=20120327-19:52:32#56=TT_ORDER#
11=32427030#37=022XBJ002#38=100#40=2#44=140625#47=A#54=1#55=ES#
60=20051205-09:11:59#200=201206#167=FUT#204=0#207=CME#10=037#
```

Next, TT FIX Adapter sends **Execution Report (8)** messages and then a full fill.

Response:

```
8=FIX.4.2#9=00378#35=8#49=TT_ORDER#56=REMOTE#50=TTORDPS001001#57=NONE#
34=20#52=20120327-19:52:32.120#55=ES#48=00A0FM00ESZ#10455=ESM2#167=FUT#
207=CME#15=USD#1=ps001001#47=A#204=0#10553=REMOTE#11=32427030#
18203=CME#18216=P15000#41=445891195#37=022XBJ002#17=022XBJ002:6#
198=QZ1Z#200=201206#151=100#14=0#54=1#40=2#77=O#59=0#11028=N#150=5#
20=0#39=5#442=1#44=140625#38=100#6=0#60=20120327-19:52:32.061#10=222#
```

Response:

```
8=FIX.4.2+9=00490+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=25+52=20120327-19:57:18.384+55=ES+48=00A0FM00ESZ+10455=ESM2+167=FUT+
207=CME+15=USD+1=ps001001+47=A+204=0+10553=REMOTE+11=32427030+
375=CME000A+18203=CME+18216=P15000+37=022XBJ002+17=1u6jv4q1fy3fu2+
58=Fi11+198=QZ1Z+10527=7133:1152775TN0032838+16018=ml1cw0+200=201206+
32=100+151=0+14=100+75=20120327+54=1+40=2+77=0+59=0+11028=N+150=2+20=0+
39=2+442=1+44=140625+38=100+31=140625+6=140625+
60=20120327-19:57:18.321+6038=20120327-19:52:32.042+10=234+
```

Canceling an order

In this conversation, a FIX client sends a **New Order Single (D)** message to buy 10 CME,FUT,ES,Jun12 contracts at 130000. TT FIX Adapter responds with an **Execution Report (8)** message confirming the order.

Request:

```
8=FIX.4.2+9=168+35=D+34=34+49=REMOTE+52=20120327-20:06:08+56=TT_ORDER+
1=ps001001+11=87554152+38=10+40=2+44=130000+47=A+54=1+55=ES+
60=20051205-09:11:59+200=201206+167=FUT+204=0+207=CME+10=187+
```

Response:

```
8=FIX.4.2+9=00363+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+
57=NONE+34=34+52=20120327-20:06:08.849+55=ES+48=00A0FM00ESZ+10455=ESM2+
167=FUT+207=CME+15=USD+1=ps001001+47=A+204=0+10553=REMOTE+11=87554152+
18203=CME+18216=P15000+37=022XKT001+17=022XKT001:0+198=QZ21+200=201206+
151=10+14=0+54=1+40=2+77=0+59=0+11028=N+150=0+20=0+39=0+442=1+
h44=130000+38=10+6=0+60=20120327-20:06:08.776+10=253+
```

The FIX client then sends a **Order Cancel Request (F)** message with Tag 11 (**clOrdID**) to cancel the order. TT FIX Adapter sends an **Execution Report (8)** to with Tag 150 (**ExecType**) set to 4 (cancel).

Request:

```
8=FIX.4.2+9=112+35=F+34=36+49=REMOTE+52=20120327-20:07:56+56=TT_ORDER+
1=ps001001+11=569732556+37=022XKT001+60=20051205-09:15:50+10=146+
```

Response:

```
8=FIX.4.2+9=00375+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=36+52=20120327-20:07:57.051+55=ES+48=00A0FM00ESZ+10455=ESM2+
167=FUT+207=CME+15=USD+1=ps001001+47=A+204=0+10553=REMOTE+11=569732556+
18203=CME+18216=P15000+41=87554152+37=022XKT001+17=022XKT001:1+
198=QZ21+200=201206+151=0+14=0+54=1+40=2+77=0+59=0+11028=N+150=4+20=0+
39=4+442=1+44=130000+38=10+6=0+60=20120327-20:07:56.977+10=090+
```

Canceling a filled order

In this conversation, a FIX client sends a **New Order Single (D)** message to buy 10 CME,FUT,ES,Jun12 contracts at 140575. TT FIX Adapter responds with an **Execution Report (8)** message confirming the order.

Request:

```
8=FIX.4.2+9=169+35=D+34=62+49=REMOTE+52=20120327-20:33:19+56=TT_ORDER+
1=ps001001+11=191983439+38=10+40=2+44=140575+47=A+54=1+55=ES+
60=20051205-09:11:59+200=201206+167=FUT+204=0+207=CME+10=007+
```

Response:

```
8=FIX.4.2+9=00364+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=62+52=20120327-20:33:19.307+55=ES+48=00A0FM00ESZ+10455=ESM2+167=FUT+
207=CME+15=USD+1=ps001001+47=A+204=0+10553=REMOTE+11=191983439+
18203=CME+18216=P15000+37=022XKT002+17=022XKT002:0+198=QZ23+200=201206+
151=10+14=0+54=1+40=2+77=0+59=0+11028=N+150=0+20=0+39=0+442=1+
44=140625+38=10+6=0+60=20120327-20:33:19.224+10=056+
```

Later, TT FIX Adapter responds with an **Execution Report (8)** message with Tag 58 (**Text**) indicating that order was filled.

Response:

```
8=FIX.4.2+9=00488+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=63+52=20120327-20:33:19.463+55=ES+48=00A0FM00ESZ+10455=ESM2+
167=FUT+207=CME+15=USD+1=ps001001+47=A+204=0+10553=REMOTE+11=191983439+
375=CME000A+18203=CME+18216=P15000+37=022XKT002+17=1uw3eri182ehnk+
58=Fill+198=QZ23+10527=7133:1169650TN0000038+16018=mlk7k0+200=201206+
32=10+151=0+14=10+75=20120328+54=1+40=2+77=0+59=0+11028=N+150=2+20=0+
39=2+442=1+44=140625+38=10+31=140500+6=140500+60=20120327-20:33:19.224+
6038=20120327-20:33:19.213+10=074+
```

Subsequently, the FIX client sends a **Order Cancel Request (F)** message cancel the order. TT FIX Adapter sends a **Order Cancel Reject (9)** message to reject the request because the order is no longer in the market.

Request:

```
8=FIX.4.2+9=112+35=F+34=63+49=REMOTE+52=20120327-20:33:55+56=TT_ORDER+
1=ps001001+11=557444141+37=022XKT002+60=20051205-09:15:50+10=132+
```

Response:

```
8=FIX.4.2+9=00205+35=9+49=TT_ORDER+56=REMOTE+57=NONE+50=NONE+34=64+
52=20120327-20:33:55.744+1=ps001001+11=557444141+37=022XKT002+
58=Order is not in the market+198=QZ23+10553=REMOTE+102=0+434=1+
39=2+60=20120327-20:33:55.744+10=017+
```

Changing a filled order

In this conversation, a FIX client submits a new order to the market and then tries to modify it after the order was filled. The FIX client sends a **New Order Single (D)** message to buy 10 CBOT,FUT,ZB,Jun12 contracts at 140625. TT FIX Adapter responds with an **Execution Report (8)** message confirming the order.

Request:

```
8=FIX.4.2+9=169+35=D+34=1533+49=REMOTE+52=20120328-20:59:01+
56=TT_ORDER+1=ps001001+11=1748690+38=10+40=2+44=140575+47=A+54=1+55=ES+
60=20051205-09:11:59+200=201206+167=FUT+204=0+207=CME+10=003+
```

Response :

```
8=FIX.4.2+9=00364+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=1561+52=20120328-20:59:01.168+55=ES+48=00A0FM00ESZ+10455=ESM2+
167=FUT+207=CME+15=USD+1=ps001001+47=A+204=0+10553=REMOTE+11=1748690+
18203=CME+18216=P15000+37=022XKT012+17=022XKT012:0+198=QZ4A+200=201206+
151=10+14=0+54=1+40=2+77=0+59=0+11028=N+150=0+20=0+39=0+442=1+
44=140575+38=10+6=0+60=20120328-20:59:01.113+10=073+
```

Response :

```
8=FIX.4.2+9=00487+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=1562+52=20120328-20:59:01.309+55=ES+48=00A0FM00ESZ+10455=ESM2+
167=FUT+207=CME+15=USD+1=ps001001+47=A+204=0+10553=REMOTE+11=1748690+
375=CME000A+18203=CME+18216=P15000+37=022XKT012+17=1cs4uawhg9h60+
58=Fill+198=QZ4A+10527=7133:1589234TN0000028+16018=mlm280+200=201206+
32=10+151=0+14=10+75=20120329+54=1+40=2+77=0+59=0+11028=N+150=2+20=0+
39=2+442=1+44=140575+38=10+31=139725+6=139725+60=20120328-20:59:01.113+
6038=20120328-20:59:01.074+10=213+
```

Subsequently, the FIX client sends a **Order Cancel Replace Request (G)** message change the price of the order to 140500. TT FIX Adapter sends a **Order Cancel Reject (9)** message to reject the request because the order is no longer in the market.

Request :

```
8=FIX.4.2+9=173+35=G+34=1538+49=REMOTE+52=20120328-21:03:56+
56=TT_ORDER+11=563239356+37=022XKT012+38=10+40=2+44=140500+47=A+54=1+
55=ES+60=20051205-09:11:59+200=201206+167=FUT+204=0+207=CME+10=179+
```

Response :

```
8=FIX.4.2+9=00196+35=9+49=TT_ORDER+56=REMOTE+57=NONE+50=NONE+34=1567+
52=20120328-21:03:56.323+11=563239356+37=022XKT012+58=Order is not in
the market+198=QZ4A+10553=REMOTE+102=0+434=2+39=2+
60=20120328-21:03:56.323+10=026+
```

Positions and fills

Overview

These examples show how TT FIX Adapter responds to a **Request For Position (UAN)** message in the following scenarios:

- [Requesting positions](#)
- [Position report subscriptions through TT Gateway roll-overs](#)
- [Requesting historical fills](#)
- [Requesting positions when none exist](#)
- [Requesting historical fills when none exist](#)
- [Receiving fill updates from a fill confirm](#)
- [Receiving fill updates from OM gateways](#)

Requesting positions

In this conversation, a FIX client sends a **Request For Position (UAN)** message with Tag 16724 (**PosReqType**) = 0 (positions) to request positions. It also provides Tag 1 (**Account**) to filter the positions for a single account.

Request

```
FA_order8=FIX.4.2+9=84+35=UAN+34=55+49=REMOTE+52=20120404-20:21:08+56=TT_ORDER+1=ps001001+16710=11+16724=0+10=195+
```

TT FIX Adapter responds by sending a **Position Report (UAP)** message for each contract in which the specified account has a position or realized P&L.

Response :

```
FA_order8=FIX.4.2+9=00250+35=UAP+49=TT_ORDER+56=REMOTE+50=TTORDPS001+57=NONE+34=77+52=20120404-20:21:08.124+16710=11+16721=11:1+58=Position+16018=mlx680+55=ES+48=00A0FM00ESZ+10455=ESM2+167=FUT+207=CMB+15=USD+1=ps001001+16727=5+200=201206+32=-1+16724=0+16210=0+31=140300+146=0+10=001+
```

Response :

```
FA_order8=FIX.4.2+9=00260+35=UAP+49=TT_ORDER+56=REMOTE+50=TTORDPS001+57=NONE+34=78+52=20120404-20:21:08.124+16710=11+16721=11:2+58=Position+16018=mlx900+55=ZB+48=00A0FM00ZBZ+10455=ZBM2+167=FUT+207=CBOT+15=USD+1=ps001001+16727=5+200=201206+32=6+16724=0+16210=0+31=137.604166666666+146=0+10=082+
```

Response :

```
FA_order8=FIX.4.2+9=00249+35=UAP+49=TT_ORDER+56=REMOTE+50=TTORDPS001+57=NONE+34=79+52=20120404-20:21:08.124+16710=11+16721=11:3+58=Position+16018=mlx680+55=6A+48=00A0IM006AZ+10455=6AU2+167=FUT+207=CMB+15=USD+1=ps001001+16727=5+200=201209+32=-5+16724=0+16210=0+31=10089+146=0+10=150+
```

Response :

```
FA_order8=FIX.4.2+9=00255+35=UAP+49=TT_ORDER+56=REMOTE+50=TTORDPS001+57
=NONE+
34=80+52=20120404-20:21:08.124+16710=11+16721=11:4+58=Position+
16018=m1x680+55=ES+48=00A0LM00ESZ+10455=ESZ2+167=FUT+207=CME+15=USD+
1=ps001001+16727=5+200=201212+32=-32+16724=0+16210=0+31=139059.375+
146=0+10=039+
```

Response

```
FA_order8=FIX.4.2+9=00253+35=UAP+49=TT_ORDER+56=REMOTE+50=TTORDPS001+57
=NONE+
34=81+52=20120404-20:21:08.124+16710=11+16721=11:5+58=Position+
16018=m1xbs0+55=FDAX+48=FDAX062012+167=FUT+207=Eurex+15=EUR+1=ps001001+
16727=5+200=201206+32=-13+16724=0+16210=0+31=7068.26923076923+
146=0+10=188+
```

Position report subscriptions through TT Gateway roll-overs

This conversation shows how TT FIX Adapter updates positions via **Position Report (UAP)** messages when a TT Gateway rolls sessions. To receive **Position Report (UAP)** messages, the FIX client must send TT FIX Adapter a **Request For Position (UAN)** message. In this example, the FIX client requests a subscription (Tag 263 (**SubscriptionRequestType**) = 1) for positions (Tag 16724 (**PosReqType**) = 0 (Positions)).

Request :

```
8=FIX.4.2+9=78+35=UAN+34=12+49=REMOTE+52=20120426-20:13:01+
56=TT_ORDER+16710=1+16724=0+263=1+10=034+
```

In this example, there is an open position of long 70 on the CME ESJun12 contract, so TT FIX Adapter responds with a **Position Report (UAP)** message.

Response :

```
8=FIX.4.2+9=00242+35=UAP+49=TT_ORDER+56=REMOTE+50=TTORDPS001+57=NONE+
34=15+52=20120426-20:13:02.037+16710=1+16721=1:2+58=Position+
16018=m31ww0+55=ES+48=00A0FM00ESZ+10455=ESM2+167=FUT+207=CME+15=USD+
16727=11+200=201206+32=70+16724=0+16210=25.00+31=135436.071428571+
10=158+
```

When the TT Gateway rolls over, the TT Gateway will set positions and P&L to zero, which will result in a **Position Report (UAP)** message being sent from TT FIX Adapter to the FIX client.

Response :

```
8=FIX.4.2+9=00227+35=UAP+49=TT_ORDER+56=REMOTE+50=TTORDPS001+57=NONE+
34=41+52=20120426-20:23:16.382+16710=1+16721=1:13+58=Position+
16018=m33rk0+55=ES+48=00A0FM00ESZ+10455=ESM2+167=FUT+207=CME+15=USD+
16727=13+200=201206+32=0+16724=0+16210=0+31=0+10=144+
```

Requesting historical fills

In this conversation, a FIX client sends a **Request For Position (UAN)** message with Tag 16724 (**PosReqType**) = 1 (trades) to request historical fills. It also provides Tag 1 (**Account**) to filter the positions for a single account.

Request

```
FA_order8=FIX.4.2+9=83+35=UAN+34=9+49=REMOTE+52=20120404-15:18:07+56=TT_ORDER+1=ps001001+16710=12+16724=1+10=156+
```

TT FIX Adapter responds by sending a **Position Report (UAP)** message for each fill for the specified account.

Response :

```
FA_order8=FIX.4.2+9=00458+35=UAP+49=TT_ORDER+56=REMOTE+34=15+52=20120404-15:18:07.130+50=TTORDPS001001+57=NONE+55=ES+48=00A0LM00ESZ+10455=ESZ2+167=FUT+207=CME+15=USD+1=ps001001+47=A+204=0+10553=REMOTE+375=CME000A+18203=CME+18216=P15000+58=Fill+10527=7137:1030832TN0000389+16018=m1x680+200=201212+32=12+75=20120404+54=2+40=2+59=0+44=139075+38=20+31=139075+60=20120404-14:18:45.594+146=0+16710=12+16721=1214muud5edbr2+198=R0VV+37=0232N8008+17=14muud5edbr2+16727=2+442=1+16724=1+77=0+20=0+10=166+
```

Response :

```
FA_order8=FIX.4.2+9=00458+35=UAP+49=TT_ORDER+56=REMOTE+34=16+52=20120404-15:18:07.130+50=TTORDPS001001+57=NONE+55=ES+48=00A0LM00ESZ+10455=ESZ2+167=FUT+207=CME+15=USD+1=ps001001+47=A+204=0+10553=REMOTE+375=CME000A+18203=CME+18216=P15000+58=Fill+10527=7137:1034795TN0000391+16018=m1x680+200=201212+32=10+75=20120404+54=2+40=2+59=0+44=139050+38=20+31=139050+60=20120404-14:30:47.496+146=0+16710=12+16721=128wqxubb85qc4+198=R0W3+37=0232N8011+17=8wqxubb85qc4+16727=3+442=1+16724=1+77=0+20=0+10=064+
```

Response :

```
FA_order8=FIX.4.2+9=00460+35=UAP+49=TT_ORDER+56=REMOTE+34=17+52=20120404-15:18:07.130+50=TTORDPS001001+57=NONE+55=ES+48=00A0LM00ESZ+10455=ESZ2+167=FUT+207=CME+15=USD+1=ps001001+47=A+204=0+10553=REMOTE+375=CME000A+18203=CME+18216=P15000+58=Fill+10527=7137:1034863TN0000392+16018=m1x680+200=201212+32=10+75=20120404+54=2+40=2+59=0+44=139050+38=20+31=139050+60=20120404-14:31:06.330+146=0+16710=12+16721=1212jra4zj7qon3+198=R0W3+37=0232N8011+17=12jra4zj7qon3+16727=4+442=1+16724=1+77=0+20=0+10=123+
```

Response :

```
FA_order8=FIX.4.2+9=00462+35=UAP+49=TT_ORDER+56=REMOTE+34=18+52=20120404-15:18:07.130+50=TTORDPS001001+57=NONE+55=ES+48=00A0FM00ESZ+10455=ESM2+167=FUT+207=CME+15=USD+1=ps001001+47=A+204=0+10553=REMOTE+375=CME000A+18203=CME+18216=P15000+58=Fill+10527=7137:1047756TN0026380+16018=m1x680+200=201206+32=1+75=20120404+54=2+40=2+59=0+44=140300+38=1+31=140300+60=20120404-15:14:41.261+146=0+16710=12+16721=121dc1vqe1ncvwao+198=R0WH+37=0232Q0011+17=1dc1vqe1ncvwao+16727=5+442=1+16724=1+77=0+20=0+10=047+
```

Requesting positions when none exist

In this conversation, a FIX client sends a **Request For Position (UAN)** message with Tag 16724 (**PosReqType**) = 0 (positions) to request positions. It also provides Tag 1 (**Account**) to filter the positions for a single account.

Request :

```
FA_order8=FIX.4.2+9=98+35=UAN+34=4172+49=REMOTE+52=20120402-
19:01:04+56=TT_ORDER+56=TT_ORDER+1=ps001001+16710=11+16724=0+10=088+
```

Because the specified account has no positions, TT FIX Adapter sends a **Position Report (UAP)** message and sets Tag 16727 (**TotalNumPosReports**) to 0.

Response :

```
FA_order8=FIX.4.2+9=00154+35=UAP+49=TT_ORDER+56=REMOTE+50=NONE+57=NONE+
34=4172+52=20120402-19:01:04.850+16710=11+16721=11:0+58=No matching
records+1=ps001001+16727=0+16724=0+146=0+10=157+
```

Requesting historical fills when none exist

In this conversation, a FIX client sends a **Request For Position (UAN)** message with Tag 16724 (**PosReqType**) = 1 (trades) to request historical fills. It also provides Tag 1 (**Account**) to filter the positions for a single account.

Request :

```
FA_order8=FIX.4.2+9=86+35=UAN+34=4175+49=REMOTE+52=20120402-
19:03:10+56=TT_ORDER+1=ps001001+16710=12+16724=1+10=045+
```

Because the specified account has no fills, TT FIX Adapter sends a **Position Report (UAP)** message and sets Tag 16727 (**TotalNumPosReports**) to 0.

Response :

```
FA_order8=FIX.4.2+9=00154+35=UAP+49=TT_ORDER+56=REMOTE+50=NONE+57=NONE+
34=4175+52=20120402-19:03:11.068+16710=12+16721=12:0+58=No matching
records+1=ps001001+16727=0+16724=1+146=0+10=164+
```

Receiving fill updates from a fill confirm

This conversation shows how TT FIX Adapter sends out an update to a fill that has been confirmed when **Send Fill Updates** is enabled in FACT. The original fill **Execution Report (8)** message is sent by TT FIX Adapter to the FIX client:

Request :

```
8=FIX.4.2+9=00447+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=179+52=20120426-18:42:57.731+55=ES+48=ES0112060000000NN+
10455=JUN12+167=FUT+207=TTSIM+15=USD+1=ps001001+18205=A1+10553=REMOTE+
375=SIMLIVEEXC0+18203=TTSIM+37=023DBV012+17=Q41US002PC2V+58=Fill+
198=1542192+10527=2928735+16018=m335c0+200=201206+32=1+151=0+14=1+54=1+
40=2+77=0+59=0+11028=Y+150=2+20=0+39=2+442=1+44=139700+38=1+31=139700+
6=139700+60=20120426-13:44:24.748+6038=20120426-18:42:57.291+10=185+
```

In X_TRADER, the account number is changed from *ps001001* to *ps001001_conf* and the fill is confirmed. TT FIX Adapter sends the following fill update **Execution Report (8)** message to the FIX client with Tag 20 (**ExecTransType**) = 2 (Correct), a new Tag 17 (**ExecID**) value, the previous Tag 17 (**ExecID**) value in Tag 19 (**ExecRefID**), and the original Tag 17 (**ExecID**) value in Tag 16017 (**OrigExecID**). Note that since this is the first update to the fill, tag Tag 19 (**ExecRefID**) and Tag 16017 (**OrigExecID**) have the same value.

Response :

```
8=FIX.4.2+9=00519+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=181+52=20120426-18:43:58.837+55=ES+48=ES0112060000000NN+
10455=JUN12+167=FUT+207=TTSIM+15=USD+1=ps001001_conf+18205=A1+
10553=REMOTE+375=SIMLIVEEXC0+18203=TTSIM+37=023DBV012+
17=Q41US002PC2V:1+58=Clearing Account now ps001001_conf+198=1542192+
10527=2928735+19=Q41US002PC2V+16017=Q41US002PC2V+16018=m335c0+
200=201206+32=1+151=0+14=1+54=1+40=2+77=0+59=0+11028=Y+150=2+20=2+39=2+
442=1+44=139700+38=1+31=139700+6=139700+60=20120426-13:44:24.748+
6038=20120426-18:42:57.291+10=081+
```

Receiving fill updates from OM gateways

This conversation shows how TT FIX Adapter sends out an update to a fill due to an OM Gateway fill update when **Send Fill Updates** is enabled in FACT, in this case from a TT OSE Gateway. The original fill **Execution Report (8)** is sent by TT FIX Adapter to the FIX client:

Response :

```
8=FIX.4.2+9=00434+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=212+52=20120426-19:01:58.064+55=DJIA+48=FUT_DJIA_1209+
10455=FUT_DJIA_1209+167=FUT+207=OSE+15=JPY+1=ps001001+18205=A1+
10553=REMOTE+18203=OSE+37=023DBV043+17=NRQ30010B14N200G9K0000001+
58=Fill+198=4BA7F40A:A6E9A5C9+16018=m32oo0+200=201209+32=1+151=0+14=1+
54=2+40=2+77=0+59=0+11028=Y+150=2+20=0+39=2+442=1+44=12500+38=1+
31=12500+6=12500+60=20120426-19:01:57+6038=20120426-19:01:57.283+
10=238+
```

The TT OSE Gateway sends a fill update to provide additional clearing information, in this case Tag 10527 (**SecondaryExecID**). TT FIX Adapter sends the following fill update **Execution Report (8)** to the FIX client with Tag 20 (**ExecTransType**) = 2 (Correct), a new Tag 17 (**ExecID**) value, the previous Tag 17 (**ExecID**) value in Tag 19 (**ExecRefID**), and the original Tag 17 (**ExecID**) value in Tag 16017 (**OrigExecID**), and the newly provided Tag 10527 (**SecondaryExecID**). Note that since this is the first update to the fill, Tag 19 (**ExecRefID**) and tag Tag 16017 (**OrigExecID**) have the same value.

Response :

```
8=FIX.4.2+9=00539+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=213+52=20120426-19:01:58.074+55=DJIA+48=FUT_DJIA_1209+
10455=FUT_DJIA_1209+167=FUT+207=OSE+15=JPY+1=ps001001+18205=A1+
10553=REMOTE+18203=OSE+37=023DBV043+17=NRQ30010B14N200G9K0000001:2+
58=Fee or Clearing Update+198=4BA7F40A:A6E9A5C9+10527=51848+
19=NRQ30010B14N200G9K0000001+16017=NRQ30010B14N200G9K0000001+
16018=m32oo0+200=201209+32=1+151=0+14=1+75=20120427+54=2+40=2+77=0+
59=0+11028=Y+150=2+20=2+39=2+442=1+44=12500+38=1+31=12500+6=12500+
60=20120426-19:01:57+6038=20120426-19:01:57.283+10=046+
```

If that fill were then confirmed in X_TRADER, an additional fill update would be sent. In X_TRADER, the account number is changed from *ps001001* to *ps001001_conf* and the fill is confirmed. TT FIX Adapter sends the following fill update **Execution Report (8)** to the FIX client with Tag 20 (**ExecTransType**) = 2 (Correct), a new Tag 17 (**ExecID**) value, the previous Tag 17 (**ExecID**) value in Tag 19 (**ExecRefID**), and the original Tag 17 (**ExecID**) value in Tag 16017 (**OrigExecID**). Note that since this is not the first update to the fill, Tag 19 (**ExecRefID**) and Tag 16017 (**OrigExecID**) now have different values.

Response :

```
8=FIX.4.2+9=00582+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=214+52=20120426-19:15:53.892+55=DJIA+48=FUT_DJIA_1209+
10455=FUT_DJIA_1209+167=FUT+207=OSE+15=JPY+1=ps001001_conf+18205=A1+
10553=REMOTE+18203=OSE+37=023DBV043+17=NRQ30010B14N200G9K0000001:3+
58=Clearing Account now ps001001_conf, Fee or Clearing Update+
198=4BA7F40A:A6E9A5C9+10527=51848+19=NRQ30010B14N200G9K0000001:2+
16017=NRQ30010B14N200G9K0000001+16018=m32oo0+200=201209+32=1+151=0+
14=1+75=20120427+54=2+40=2+77=O+59=0+11028=Y+150=2+20=2+39=2+442=1+
44=12500+38=1+31=12500+6=12500+60=20120426-19:01:57+
6038=20120426-19:01:57.283+10=163+
```

Order status

Overview

These examples show how TT FIX Adapter responds to an **Order Status Request** (H) message in the following scenarios:

- [Request an order book download](#)
- [Request a filtered order book download](#)
- [Request order status with no orders in the market](#)

Request an order book download

In this conversation, a FIX client sends an **Order Status Request** (H) message to TT FIX Adapter. Because the message omits Tags 11 (C1OrdID) and 37 (OrderID), TT FIX Adapter treats it as a request for an order book download.

Request :

```
8=FIX.4.2+9=56+35=H+34=1548+49=REMOTE+52=20120328-21:12:06+
56=TT_ORDER+10=013+
```

TT FIX Adapter responds with an **Execution Report** (8) for each working order visible to the FIX client. In this example, Tag 16728 (TotalNumOrders) indicates the order book contains 33 working orders. However, the example shows the report for only the first order.

Response :

```
8=FIX.4.2+9=00390+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=1577+52=20120328-21:12:06.429+55=ES+48=ES0112030000000NN+
10455=MAR12+167=FUT+207=CME+15=USD+1=ps001001+47=A+204=0+10553=TRADERA+
18203=CME+37=022XZA016+17=022XZA016:2+58=Order Book Download+
198=1300913+200=201203+151=8+14=2+16728=33+54=1+40=2+77=O+59=0+11028=Y+
150=D+20=3+39=1+442=1+378=4+44=50+38=10+6=50+60=20120328-21:12:06.429+
10=205+
```

Request a filtered order book download

In this conversation, a FIX client sends an **Order Status Request** (H) message to TT FIX Adapter. Because the message omits Tags 11 (C1OrdID) and 37 (OrderID), TT FIX Adapter treats it as a request for an order book download. However, in this example, the request uses Tag 1 (Account) to return only orders for a single account ID.

Request :

```
8=FIX.4.2+9=56+35=H+34=1548+1=ps001001+49=REMOTE+52=20120328-21:12:06+
56=TT_ORDER+10=013+
```

TT FIX Adapter responds with an **Execution Report** (8) for each working order visible to the FIX client where the account is ps001001. In this example, Tag 16728 (TotalNumOrders) indicates the order book for this account contains 30 working orders. However, the example shows the report for only the first order.

Response :

```
8=FIX.4.2+9=00390+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=1577+52=20120328-21:12:06.429+55=ES+48=ES0112030000000NN+
10455=MAR12+167=FUT+207=TTSIM+15=USD+1=ps001001+47=A+204=0+
10553=TRADERA+18203=CME+37=022XZA016+17=022XZA016:2+
58=Order Book Download+198=1300913+200=201203+151=8+14=2+16728=30+54=1+
40=2+77=O+59=0+11028=Y+150=D+20=3+39=1+442=1+378=4+44=50+38=10+6=50+
60=20120328-21:12:06.429+10=205+
```

Request order status with no orders in the market

In this conversation, a FIX client sends an **Order Status Request (H)** message to TT FIX Adapter. Because the message omits Tags 11 (**clOrdID**) and 37 (**OrderID**), TT FIX Adapter treats it as a request for an order book download.

Request :

```
8=FIX.4.2+9=56+35=H+34=1543+49=REMOTE+52=20120328-
21:07:59+56=TT_ORDER+10=020+
```

In this example, the FIX client has no working orders, so TT FIX Adapter responds with an **Execution Report (8)** with Tag 150 (**ExecType**)=8 and Tag 16728 (**TotalNumOrders**)=0.

Response :

```
8=FIX.4.2+9=00390+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+
57=NONE+34=1572+52=20120328-21:07:59.930+55=ES+48=ES01120300000000NN+
10455=MAR12+167=FUT+207=TTSIM+15=USD+1=ps001001+47=A+204=0+
10553=REMOTE+18203=TTSIM+37=022XZA016+17=022XZA016:1+
58=Order Book Download+198=1300913+200=201203+151=8+14=2+16728=1+
54=1+40=2+77=0+59=0+11028=Y+150=D+20=3+39=1+442=1+378=4+44=50+38=10+
6=50+60=20120328-21:07:59.930+10=217+
```

Gateway status

Overview	The following example shows how TT FIX Adapter responds to a Gateway Status Request (UAR) message.
Subscribing for gateway status updates	In this conversation, a FIX client sends a Gateway Status Request (UAR) message to subscribe for gateway status updates by setting Tag 263 (SubscriptionRequestType) to 1 (snapshot and updates).

Request :

```
8=FIX.4.2+9=85+35=UAR+34=42+49=REMOTE+52=20120329-
21:24:08+56=TT_ORDER+56=TT_ORDER+263=1+18200=ALL+10=177+
```

TT FIX Adapter responds with a **Gateway Status (UAT)** message (snapshot) that indicates the current status of the Price Server and all servers on all gateways to which TT FIX Adapter connects.

Response :

```
8=FIX.4.2+9=00589+35=UAT+49=TT_ORDER+56=REMOTE+34=50+
52=20120329-21:24:08.518+18200=ALL+18201=12+18202=2+207=CBOT+
18203=CBOT+18204=1+18202=2+207=CBOT+18203=CBOT+18204=2+18202=2+
207=CBOT+18203=CBOT+18204=3+18202=2+207=CME+18203=CME+18204=1+
18202=2+207=CME+18203=CME+18204=2+18202=2+207=CME+18203=CME+18204=3+
18202=2+207=Eurex+18203=Eurex+18204=1+18202=2+207=Eurex+18203=Eurex+
18204=2+18202=2+207=Eurex+18203=Eurex+18204=3+18202=2+207=SSE+
18203=SSE-B+18204=2+18202=2+207=SSE+18203=SSE-B+18204=3+10=135+
```

At this point, the CBOT Price Server is stopped. TT FIX Adapter sends a **Gateway Status (UAT)** message (update) with 18202=3 to indicate this situation.

Response :

```
8=FIX.4.2+9=00120+35=UAT+49=TT_ORDER+56=REMOTE+34=69+
52=20120329-21:39:46.372+18200=ALL+325=Y+18201=1+18202=3+207=CBOT+
18203=CBOT+18204=1+10=153+
```

At this point, the Price Server restarts. TT FIX Adapter sends an update with 18202=4 to indicate this situation.

Response :

```
8=FIX.4.2+9=00120+35=UAT+49=TT_ORDER+56=REMOTE+34=73+
52=20120329-21:40:00.778+18200=ALL+325=Y+18201=1+18202=4+207=CBOT+
18203=CBOT+18204=1+10=141+
```

When the Price Server completes its restart, TT FIX Adapter sends an update with 18202=2 to indicate this situation.

Response :

```
8=FIX.4.2+9=00120+35=UAT+49=TT_ORDER+56=REMOTE+34=76+
52=20120329-21:40:01.418+18200=ALL+325=Y+18201=1+18202=2+207=CBOT+
18203=CBOT+18204=1+10=134+
```

Staged Orders

Overview

These examples show how TT FIX Adapter processes a staged order in the following scenarios:

- [Staged Order Submitted, Claimed, and Filled](#)
- [Staged Order Submitted, Claimed, and then Unclaimed](#)
- [Staged Order Submitted and then Deleted](#)

Staged Order Submitted, Claimed, and Filled

This conversation shows a staged order being submitted, then claimed by an X_TRADER user who submits child orders that get filled until the staged order itself is fully filled.

First, the FIX client sends a 100 lot buy internal staged order.

Request :

```
8=FIX.4.2+9=212+35=D+34=86+49=REMOTE+52=20120426-21:25:32+56=TT_ORDER+
1=ps001001+11=321868110+38=100+40=2+21=3+44=139150+47=A+54=1+55=ES+
60=20051205-09:11:59+200=201206+167=FUT+204=0+207=TTSIM+
16106=StagedOrderMsg+16111=I+10=005+
```

TT FIX Adapter sends an **Execution Report (8)** message confirming the order.

Response :

```
8=FIX.4.2+9=00481+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=108+52=20120426-21:25:32.070+55=ES+48=ES0112060000000NN+
10455=JUN12+167=FUT+207=TTSIM+15=USD+1=ps001001+47=A+204=0+
10553=REMOTE+11=321868110+18203=TTSIM+16106=StagedOrderMsg+
18208=SSE-B+37=OSD88X005+17=OSD88X005:0+
58=From Gateway: Staged Order: Buy 100, LIM 139150, GTD+
198=1002688+200=201206+151=100+14=0+54=1+40=2+77=0+59=0+21=3+16109=A+
16111=I+11028=N+150=0+20=0+39=0+442=1+44=139150+38=100+6=0+
60=20120426-21:24:30+10=204+
```

The X_TRADER user *EXECTRADER* claims the staged order. TT FIX Adapter sends an **Execution Report (8)** message confirming that *EXECTRADER* claimed the order.

Response :

```
8=FIX.4.2+9=00502+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=109+52=20120426-21:25:36.234+55=ES+48=ES0112060000000NN+
10455=JUN12+167=FUT+207=TTSIM+15=USD+1=ps001001+47=A+204=0+
10553=REMOTE+11=321868110+18203=TTSIM+16106=StagedOrderMsg+
16110=EXECTRADER+18208=SSE-A+37=OSD88X005+17=OSD88X005:1+
58=From Gateway: Staged Order OSD88X005 claimed by EXECTRADER+
198=1002688+200=201206+151=100+14=0+54=1+40=2+77=0+59=0+21=3+16109=O+
16111=I+11028=N+150=D+20=0+39=5+442=1+378=4+44=139150+38=100+6=0+
60=20120426-21:25:36.234+10=239+
```

The X_TRADER user *EXECTRADER* submits two 50 lot child orders that get filled. Note that fill updates are sent on both the child order and the parent staged order.

TT FIX Adapter sends an **Execution Report (8)** message confirming the first 50 lot child order.

Response :

```
8=FIX.4.2+9=00409+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=110+52=20120426-21:25:43.577+55=ES+48=ES0112060000000NN+
10455=JUN12+167=FUT+207=TTSIM+15=USD+1=ps001001+47=A+204=0+
10553=EXECTRADER+18203=TTSIM+37=023DBV051+17=023DBV051:0+
58=Created from existing+198=1640257+200=201206+151=50+14=0+54=1+40=2+
77=0+59=0+11028=Y+150=0+20=0+39=0+442=1+44=139150+38=50+6=0+
60=20120426-21:25:43.577+16112=1+16113=0SD88X005+16114=7+10=246+
```

When the first child order is filled, TT FIX Adapter sends an **Execution Report (8)** message for the child order.

Response :

```
8=FIX.4.2+9=00481+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=112+52=20120426-21:25:45+55=ES+48=ES0112060000000NN+10455=JUN12+
167=FUT+207=TTSIM+15=USD+1=ps001001+47=A+204=0+10553=EXECTRADER+
375=SIMLIVEEXC0+18203=TTSIM+37=023DBV051+17=Q41US002PFLP+58=Fill+
198=1640257+10527=2932409+16018=m335c0+200=201206+32=17+151=0+14=50+
54=1+40=2+77=0+59=0+11028=Y+150=1+20=0+39=1+442=1+44=139150+38=50+
31=139150+6=139150+60=20120426-16:27:12.545+6038=20120426-21:24:41.947+
16112=1+16113=0SD88X005+16114=7+10=047+
```

TT FIX Adapter also sends a partial fill **Execution Report (8)** message for the parent staged order.

Response :

```
8=FIX.4.2+9=00500+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=8+52=20120516-19:28:25.273+55=ES+48=ES0112060000000NN+
10455=JUN12+167=FUT+207=TTSIM+15=USD+1=ps001001+47=A+204=0+
10553=REMOTE+11=589549387+18203=TTSIM+16106=StagedOrderMsg+
16110=EXECTRADER+18208=SSE-A+37=023QEH001+17=SSE-A4fb3f455+58=Fill+
198=1000342+16018=m439c0+200=201206+32=50+151=50+14=50+54=1+40=2+77=0+
59=0+21=3+16109=0+16111=I+11028=N+150=1+20=0+39=1+442=1+44=139150+
38=100+31=132500+6=132500+60=20120516-19:27:46+
6038=20120516-19:26:30.899+10=119
```


TT FIX Adapter sends an **Execution Report (8)** message confirming the second 50 lot child order.

Response :

```
8=FIX.4.2+9=00409+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=119+52=20120426-21:25:52.485+55=ES+48=ES0112060000000NN+
10455=JUN12+167=FUT+207=TTSIM+15=USD+1=ps001001+47=A+204=0+
10553=EXEctrader+18203=TTSIM+37=023DBV052+17=023DBV052:0+
58=Created from existing+198=1640313+200=201206+151=50+14=0+54=1+40=2+
77=0+59=0+11028=Y+150=0+20=0+39=0+442=1+44=139150+38=50+6=0+
60=20120426-21:25:52.485+16112=1+16113=0SD88X005+16114=7+10=246+
```

When the first child order is filled, TT FIX Adapter sends an **Execution Report (8)** message for the child order.

Response :

```
8=FIX.4.2+9=00485+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=120+52=20120426-21:25:52.485+55=ES+48=ES0112060000000NN+
10455=JUN12+167=FUT+207=TTSIM+15=USD+1=ps001001+47=A+204=0+
10553=EXEctrader+375=SIMLIVEEXC0+18203=TTSIM+37=023DBV052+
17=Q41US002PFLV+58=Fill+198=1640313+10527=2932415+
16018=m335c0+200=201206+32=27+151=23+14=27+54=1+40=2+77=0+59=0+11028=Y+
150=1+20=0+39=1+442=1+44=139150+38=50+31=139150+6=139150+
60=20120426-16:27:20.045+6038=20120426-21:24:50.851+16112=1+
16113=0SD88X005+16114=7+10=241+
```

Finally, TT FIX Adapter sends the parent staged order fill.

Response :

```
8=FIX.4.2+9=00508+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=122+52=20120426-21:25:52.505+55=ES+48=ES0112060000000NN+
10455=JUN12+167=FUT+207=TTSIM+15=USD+1=ps001001+47=A+204=0+
10553=REMOTE+11=321868110+18203=TTSIM+16106=StagedOrderMsg+
16110=EXEctrader+18208=SSE-A+37=0SD88X005+17=SSE-B4f9245d9+58=Fill+
198=1002688+16018=m32800+200=201206+32=100+151=0+14=100+54=1+40=2+77=0+
59=0+21=3+16109=O+16111=I+11028=N+150=2+20=0+39=2+442=1+44=139150+
38=100+31=139150+6=139150+60=20120426-21:24:50+
6038=20120426-21:24:34.600+10=097+
```

Staged Order Submitted, Claimed, and then Unclaimed

This conversation shows a staged order being submitted, claimed by an X_TRADER user, and then unclaimed by the X_TRADER user.

First, the FIX client sends a 100 lot buy staged order.

Request :

```
8=FIX.4.2+9=212+35=D+34=107+49=REMOTE+52=20120426-21:45:04+
56=TT_ORDER+1=ps001001+11=44678303+38=100+40=2+21=3+44=139150+47=A+
54=1+55=ES+60=20051205-09:11:59+200=201206+167=FUT+204=0+207=TTSIM+
16106=StagedOrderMsg+16111=I+10=005+
```

TT FIX Adapter sends an **Execution Report (8)** confirming the order.

Response :

```
8=FIX.4.2+9=00480+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=147+52=20120426-21:45:04.527+55=ES+48=ES0112060000000NN+
10455=JUN12+167=FUT+207=TTSIM+15=USD+1=ps001001+47=A+204=0+
10553=REMOTE+11=44678303+18203=TTSIM+16106=StagedOrderMsg+18208=SSE-
B+37=0SD88X007+17=0SD88X007:0+
58=From Gateway: Staged Order: Buy 100, LIM 139150, GTD+198=1002692+
200=201206+151=100+14=0+54=1+40=2+77=0+59=0+21=3+16109=A+16111=I+
11028=N+150=0+20=0+39=0+442=1+44=139150+38=100+6=0+
60=20120426-21:44:02+10=171+
```

The X_TRADER user *EXECTRADER* claims the staged order. TT FIX Adapter sends an **Execution Report (8)** message confirming that *EXECTRADER* claimed the order.

Response :

```
8=FIX.4.2+9=00501+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=148+52=20120426-21:45:13.016+55=ES+48=ES0112060000000NN+
10455=JUN12+167=FUT+207=TTSIM+15=USD+1=ps001001+47=A+204=0+
10553=REMOTE+11=44678303+18203=TTSIM+16106=StagedOrderMsg+
16110=EXECTRADER+18208=SSE-B+37=0SD88X007+17=0SD88X007:1+
58=From Gateway: Staged Order 0SD88X007 claimed by EXECTRADER+
198=1002692+200=201206+151=100+14=0+54=1+40=2+77=0+59=0+21=3+16109=O+
16111=I+11028=N+150=D+20=0+39=5+442=1+378=4+44=139150+38=100+6=0+
60=20120426-21:45:13.015+10=188+
```

X_TRADER user *EXECTRADER* unclaims the staged order. TT FIX Adapter sends an **Execution Report (8)** message confirming that the order is released. The staged order is now available again to be claimed.

Response :

```
8=FIX.4.2+9=00491+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=150+52=20120426-21:46:24.644+55=ES+48=ES0112060000000NN+
10455=JUN12+167=FUT+207=TTSIM+15=USD+1=ps001001+47=A+204=0+
10553=REMOTE+11=44678303+18203=TTSIM+16106=StagedOrderMsg+
18208=SSE-B+37=0SD88X007+17=0SD88X007:2+
58=From Gateway: Staged Order 0SD88X007 released by EXECTRADER+
198=1002692+200=201206+151=100+14=0+54=1+40=2+77=0+59=0+21=3+16109=A+
16111=I+11028=N+150=D+20=0+39=5+442=1+378=4+44=139150+38=100+6=0+
60=20120426-21:46:24.644+10=199+
```

Staged Order Submitted and then Deleted

This conversation shows a staged order being submitted and then deleted. First, the FIX client sends a 100 lot buy staged order.

Request :

```
8=FIX.4.2+9=212+35=D+34=87+49=REMOTE+52=20120426-21:26:22+
56=TT_ORDER+1=ps001001+11=225643513+38=100+40=2+21=3+44=139150+47=A+
54=1+55=ES+60=20051205-09:11:59+200=201206+167=FUT+204=0+207=TTSIM+
16106=StagedOrderMsg+16111=I+10=007+
```

TT FIX Adapter sends an **Execution Report (8)** message confirming the order.

Response :

```
8=FIX.4.2+9=00481+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=123+52=20120426-21:26:22.480+55=ES+48=ES0112060000000NN+
10455=JUN12+167=FUT+207=TTSIM+15=USD+1=ps001001+47=A+204=0+
10553=REMOTE+11=225643513+18203=TTSIM+16106=StagedOrderMsg+
18208=SSE-B+37=0SD88X006+17=0SD88X006:0+58=From Gateway: Staged Order:
Buy 100, LIM 139150, GTD+198=1002691+200=201206+151=100+14=0+54=1+
40=2+77=O+59=0+21=3+16109=A+16111=I+11028=N+150=0+20=0+39=0+442=1+
44=139150+38=100+6=0+60=20120426-21:25:20+10=203+
```

The FIX client sends an **Order Cancel Request (F)**.

Request :

```
8=FIX.4.2+9=112+35=F+34=88+49=REMOTE+52=20120426-21:26:56+56=TT_ORDER+
1=ps001001+37=0SD88X006+60=20051205-09:15:50+11=562767830+10=160+
```

TT FIX Adapter sends a **Pending Cancel (39=6) Execution Report (8)** followed by a **Cancel (39=4) Execution Report (8)** when the cancel is confirmed.

Response :

```
8=FIX.4.2+9=00438+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=124+52=20120426-21:26:56.169+55=ES+48=ES0112060000000NN+
10455=JUN12+167=FUT+207=TTSIM+15=USD+1=ps001001+47=A+204=0+
10553=REMOTE+11=562767830+18203=TTSIM+16106=StagedOrderMsg+18208=SSE-
B+41=225643513+37=0SD88X006+17=0SD88X006:1+
198=1002691+200=201206+151=100+14=0+54=1+40=2+77=O+59=0+21=3+16109=A+
16111=I+11028=N+150=6+20=0+39=6+442=1+44=139150+38=100+6=0+
60=20120426-21:25:20+10=127+
```

Response :

```
8=FIX.4.2+9=00440+35=8+49=TT_ORDER+56=REMOTE+50=TTORDPS001001+57=NONE+
34=125+52=20120426-21:26:56.215+55=ES+48=ES0112060000000NN+
10455=JUN12+167=FUT+207=TTSIM+15=USD+1=ps001001+47=A+204=0+
10553=REMOTE+11=562767830+18203=TTSIM+16106=StagedOrderMsg+
18208=SSE-B+41=225643513+37=0SD88X006+17=0SD88X006:2+198=1002691+
200=201206+151=0+14=0+54=1+40=2+77=O+59=0+21=3+16109=A+16111=I+11028=N+
150=4+20=0+39=4+442=1+44=139150+38=100+6=0+60=20120426-21:26:56.215+
10=221+
```

Appendix overview

This appendix includes information on exchange-specific tags and values and price display codes.

In this chapter

Section	Page
Enumerations	206
Price display codes	209

Enumerations

Tag 35 (MsgType)

Tag 35 (`MsgType`) supports the following values:

ID	Name	ID	Name
0	Heartbeat	F	Order Cancel Request
1	Test Request	f	Security Status
2	Resend Request	G	Order Cancel Replace Request
3	Reject	H	Order Status Request
4	Sequence Reset	j	Business Message Reject
5	Logout	UAN	Request For Position
8	Execution Report	UAP	Position Report
9	Order Cancel Reject	UAR	Gateway Status Request
A	Logon	UAT	Gateway Status
c	Security Definition Request	V	Market Data Request
D	New Order Single	W	Market Data Snapshot Full Refresh
d	Security Definition	X	Market Data Incremental Refresh
e	Security Status Request	Y	Market Data Request Reject

Tag 58 (Text)

When an order is rejected for risk violation reasons (Tag 39=8), TT FIX Adapter sends an **Execution Report** (8) with Tag 58 (`Text`) set to one of the following strings.

String	Description
GW:ORDQEntered_Order_Qty > OQLMTMax_Order_Qty_Limit Additional_GW_specific_information	The maximum order quantity limit was exceeded.
GW:WCPComputed_Worst_Case_Position L(Long)/S(Short) > PLMTMax Position Limit	The maximum position limit was exceeded.
GW:Excds CrLmt (LMT<Max_P&L_Limit> +PLCurrent_P&L_MMargin=AVCalculation_Result) Traders_Currency Additional_GW_specific_information	The maximum P&L or margin limit was exceeded.

Tag 207 (SecurityExchange)

Tag 207 (`SecurityExchange`) supports the following values.

- BTec
- BVMF
- CBOE
- CBOT
- CFE
- CME
- EBS
- Eris_GovEx
- Eurex
- FIX
- FIX-A...FIX-Z
- ICE_IPE
- LME
- MEFF
- Montreal
- MX
- NYSE_Liffe
- NYSE_Liffe_US
- OSE
- SFE
- SGX
- TFX
- TOCOM
- TTSIM

Some exchanges might list some products on other exchanges. For example, the European Climate Exchange lists products on ICE_IPE. For those products, you must specify **ICE_IPE** in Tag 207 (**SecurityExchange**).

Note: The BM&F BOVESPA exchange lists products on both CME and BVMF. If you have a TT CME Gateway, you must specify **CME** in Tag 207 (**SecurityExchange**). If you have a TT BVMF Gateway, you must specify **BVMF** in Tag 207 (**SecurityExchange**).

Exchange Name	Tag Value	Exchange Name	Tag Value	Exchange Name	Tag Value
BM&FBOVESPA*	CME or BVMF	Euronext Brussels	NYSE_Liffe	MEFF	MEFF
BrokerTec	BTec	Euronext Lisbon	NYSE_Liffe	Minneapolis Grain Exchange	CBOT
CBOE Futures Exchange*	CBOE or CFE	Euronext Paris	NYSE_Liffe	Montreal* Climate Exchange	Montreal or MX
CCX Chicago Climate Futures Exchange	ICE_IPE	European Climate Exchange	ICE_IPE	Montreal* Exchange	Montreal or MX
CME Group - CBOT*	CBOT or CME	European Energy Exchange	ICE_IPE	NYSE Liffe	NYSE_Liffe
CME Group - CME	CME	GovEx	Eris_GovE x	NYSE Liffe US*	NYSE_Liffe _US or eCBOT
CME Group - NYMEX	CME	ICE Futures Canada	ICE_IPE	Osaka Security Exchange	OSE
Dubai Mercantile Exchange	CME	ICE Futures Europe	ICE_IPE	Singapore Exchange (SGX)	SGX
EBS	EBS	ICE Futures U.S.	ICE_IPE	Sydney Futures Exchange	SFE
Eris	Eris_GovE x	Intercontinenta l Exchange (ICE)	ICE_IPE	Tokyo Commodity Exchange (TOCOM)	TOCOM
Eurex	Eurex	Kansas City Board of Trade	CBOT	Tokyo Financial Exchange	TFX
Euronext Amsterdam	NYSE_Liffe	London Metal Exchange	LME		

* Use the value corresponding to the TT Gateway to which TT FIX Adapter connects

**Tag 10762
(SecuritySubType)**Tag 10762 (**SecuritySubType**) supports the following values:

- (empty string)
- 1x2 Ratio
- 1x3 Ratio
- 2X1 Ratio
- 2x3 Ratio
- 3-Way
- 3-Way Straddle vs. Call
- 3-Way Straddle vs. Put
- Box
- Bundle
- Butterfly
- Buy/Write
- Cabinet
- Calendar
- Call Spread vs. Put
- Misc. Call/Put Spread
- Combo
- Conditional Curve
- Condor
- Conversion/Reversal
- Covered
- Crack
- Custom Market
- Diag Calendar
- Diag Strad Calendar
- Double Butterfly
- Double
- Spread
- Guts
- Vol 1x2 Ratio
- Hedged 3-Way
- Hedged Box
- Hedged Butterfly
- Vol Calendar
- Hedged Call Spread versus Put
- Hedged Call/Put Spread
- Hedged Ladder
- Hedged Combo
- Hedged Conditional Curve
- Hedged Condor
- Conversion/Reversal vs. Short U/L
- Hedged Call/Put Diag Calendar Spread
- Hedged Diagonal Straddle Calendar Spread
- Hedged Double
- Hedged Guts
- Hedged Horizontal
- Hedged Horizontal Straddle
- Hedged Iron Butterfly
- Hedged Iron Condor
- Hedged Put Spread versus Call
- Hedged Risk Reversal
- Hedged Straddle Calendar Spread
- Hedged Straddle
- Hedged Straddle Strip
- Hedged Strangle
- Hedged Strip
- Hedged Vertical
- Hedged X-Mas Tree
- Horizontal
- Horizontal Straddle
- ICS
- Inter-Product
- Iron BFly
- Iron Condor
- Jelly Roll
- Ladder
- Month vs. Pack
- Non Standard Combination
- Option Strip
- Pack Butterfly
- Pack
- Put Spread vs. Call
- Reduced Tick Spread
- Reverse Reduced Tick Calendar Spread
- Risk Reversal
- Strad Calendar
- Straddle vs. Call/Put
- Straddle
- Straddle Strip
- Strangle
- Strip
- Synthetic Conv/Rev
- Tailor Made Combinations
- Unknown
- Vertical
- Volatility Trade
- X-Mas Tree

Price display codes

Overview TT FIX Adapter supports a variety of price display codes that allow you to display price information in different formats.

Price display codes Tag 16451 (**PriceDisplayType**) supports the following values.

Code	Description	Code	Description
0	Precision and decimal values, which is the normal Price Display.	29	N/A
1	Thirty-seconds (32nds)	30	BrokerTec 1/64: 1.00, 1.00+, 1.01, 1.01+... 1.31, 1.31+, 2.00
2	Yield price (format 1)	31	BrokerTec 2/8: 1, 1.2, 1+, 1.6, 2
3	Yield price (format 2)	32	BrokerTec 2/8 of 1/32: 1.00, 1.002, 1.00+, 1.006, 1.01.....1.31, 1.312, 1.31+, 1.316, 2
4	Discount price	33	BrokerTec 1/8 of 1/32: 1.00, 1.001, 1.002, 1.003, 1.00+, 1.005, 1.006, 1.007, 1.01
5	Decimal price	34	BrokerTec Both 1/100 and 1/8 (104 ticks per point): 1.000, 1.010, 1.020...1.120, 1.125, 1.130
6	Basis price	35	BrokerTec Normal - Uses precision and decimals values, but with series instead of prod if there is no tick table
7	Roll price	36	BrokerTec Points.FFQ FF=00.63 Q=1,+ ,3
8	Spread price	37	BrokerTec 32nds.Q, Q=1,2,3,+ ,5,6,7
9	512th price	38	Quarter thirty-seconds 4x - Price is in quarter thirty-seconds but ticks 4 at a time
10	Half-cents (last digit is either 0 or 5)	39	Quarter thirty-seconds 2x - Price is in quarter thirty-seconds but ticks 2 at a time
11	Fourths (last digit is either 0, 2, 4, or 6)	40	BrokerTec Yield - Price has 400 ticks per point, displays with 4 digits after decimal point
12	Modified fourths (last digit is either 0, 2, 5, or 7)	41	Half-cents 2x - Price is in half cents (last digit 0, 5), but ticks 2 at a time
13	Eighths	42	Normal 2x - Normal type, but tick 2 at a time
14	Modified decimal fourths (last digit is either 0, 2, 5, or 7)	43	Normal 5x - Normal type, but ticks 5 at a time
15	Thirty-seconds (Differs from 1 in that it does not use + or decimal places)	44	Normal 100x - Normal type, but ticks 100 at a time
16	Sixty-fourths (64ths)	45	Five one-thousandths, but ticks 2 at a time
17	Half thirty-seconds	46	Five one-thousandths, but ticks 5 at a time
18	Quarter thirty-seconds	47	Five one-thousandths, but ticks 10 at a time
19	Half sixty-fourths	48	Ten-hundredths of a cent, but ticks 2 at a time
20	Fives (1/20 points)	49	Ten-hundredths of a cent, but ticks 5 at a time
21	Half points	50	Ten-hundredths of a cent, but ticks 10 at a time

Table 9. Price display codes

Code	Description	Code	Description
22	Five one-thousandths	51	One-hundredths, but ticks 1/4 of 100th at a time
23	One-hundredths	52	Normal 10x - Normal type, but ticks 10 at a time
24	Tenths (dimes)	53	Normal 50x - Normal type, but ticks 50 at a time
25	Ten-hundredths of a cent	54	Normal 500x - Normal type, but ticks 500 at a time
26	No decimals	55	Normal 25x - Normal type, but ticks 25 at a time
27	Modified 1/40 points (last digit is 0, 2, 5, or 7)	56	BrokerTec inverted - Similar to BrokerTec normal, but prices are inverted
28	N/A		

Table 9. Price display codes

B

Supported FIX Messages and Tags

Appendix overview

This appendix provides lists of FIX messages and tags.

In this chapter

Section	Page
FIX messages	212
FIX tags by name	214
FIX tags by number	219

FIX messages

About the list

This section provides lists of the FIX messages supported by TT FIX Adapter sorted by ID and name.

Messages (Tag 35) listed by ID

Message ID	Message Name
0	Heartbeat
1	Test Request
2	Resend Request
3	Reject
4	Sequence Reset
5	Logout
8	Execution Report
9	Order Cancel Reject
A	Logon
c	Security Definition Request
D	New Order Single
d	Security Definition
e	Security Status Request
F	Order Cancel Request
f	Security Status
G	Order Cancel Replace Request
H	Order Status Request
j	Business Message Reject
UAN	Request For Position
UAP	Position Report
UAR	Gateway Status Request
UAT	Gateway Status
V	Market Data Request
W	Market Data Snapshot Full Refresh
X	Market Data Incremental Refresh
Y	Market Data Request Reject

Messages (Tag 35) listed by name

Message Name	Message ID
Business Message Reject	j
Execution Report	8
Gateway Status	UAT
Gateway Status Request	UAR

Message Name	Message ID
Heartbeat	0
Logon	A
Logout	5
Market Data Incremental Refresh	X
Market Data Request	V
Market Data Request Reject	Y
Market Data Snapshot Full Refresh	W
New Order Single	D
Order Cancel Reject	9
Order Cancel Replace Request	G
Order Cancel Request	F
Order Status Request	H
Position Report	UAP
Reject	3
Request For Position	UAN
Resend Request	2
Security Definition	d
Security Definition Request	c
Security Status	f
Security Status Request	e
Sequence Reset	4
Test Request	1

FIX tags by name

About this list This section provides a list of FIX tags sorted by tag name.

Tags listed by name

Tag Name	Tag #
Account	1
AggregatedBook	266
AggressorSide	16488
AutoAggressive	16481
AvgPx	6
BeginSeqNo	7
BeginString	8
Blocks	16463
BodyLength	9
BusinessRejectReason	380
BusinessRejectRefID	379
Checksum	10
ClearingAccount	440
ClOrdID	11
ContraBroker	375
ContractTerm	18211
CumQty	14
Currency	15
CustomerOrFirm	204
CxlRejReason	102
CxlRejResponseTo	434
DeliverToSubID	129
DeliveryUnit	16460
EncryptMethod	98
EndSeqNo	16
EventDate	866
EventType	865
ExactNumOrdersIndicator	16489
ExchangeGateway	18203
ExchContractSymbol	16309
ExchCred	18216
ExchPointValue	16554
ExchTickSize	16552
ExecBroker	76
ExecID	17

Table 10. FIX tags by name

Tag Name	Tag #
ExecInst	18
ExecRefID	19
ExecRestatementReason	378
ExecTransType	20
ExecType	150
ExpectedFillType	10442
ExpireDate	432
FFT2	16102
FFT3	16103
FixingDate	10541
ForceLogout	18000
GapFillFlag	123
GatewayStatus	18202
GatewayStatusReqId	18200
HandlInst	21
HeartBtInt	108
IncludeNumberOfOrders	18214
LastPx	31
LastShares	32
LeavesQty	151
LegPrice	10566
LegSide	16624
LinkID	16113
LinkType	16114
LotSize	16461
ManualOrderIndicator	11028
MarginExcess	16899
MarketDepth	264
MaturityDay	205
MaturityMonthYear	200
MaxPrice	16458
MaxShow	210
MDEntryBuyer	288
MDEntrySeller	289
MDEntryPositionNo	290
MDEntryPx	270
MDEntrySize	271
MDEntrySizeType	16487
MDEntryState	16486
MDEntryType	269

Table 10. FIX tags by name(Continued)

Tag Name	Tag #
MDReqID	262
MDUpdateAction	279
MDUpdateType	265
MinQty	110
MiscFeeAmt	137
MsgSeqNum	34
MsgType	35
MultiLegReportingType	442
NewSeqNo	36
NoEvents	864
NoGateways	18206
NoGatewayStatus	18201
NoLinks	16112
NoMDEntries	268
NoMDEntryTypes	267
NoMiscFees	136
NoRelatedSym	146
NumberOfOrders	346
NumTicks	16457
NumTickTblEntries	16456
OnBehalfOfSubID	116
OpenClose	77
OptAttribute	206
OrderEnteredTime	6038
OrderID	37
OrderOriginationID	16142
OrderQty	38
OrdRejReason	103
OrdStatus	39
OrdType	40
OrigClOrdID	41
OrigExecID	16017
OrigSendingTime	122
PassiveAggressive	16480
PointValue	16454
PosMaintRptId	16721
PosReqId	16710
PosReqType	16724
PossDupFlag	43
PossResendFlag	97

Table 10. FIX tags by name(Continued)

Tag Name	Tag #
Price	44
PriceDisplayType	16451
PriceFeedStatus	18210
PutOrCall	201
RatioQty	319
RawData	96
RealizedPnDL	16210
RefMsgType	372
RefSeqNum	45
RefTagID	371
RequestTickTable	17000
ResetSeqNumFlag	141
RoutingLevel	16111
Rule80A	47
SecondaryExecID	10527
SecondaryOrderID	198
SecurityAltID	10455
SecurityDesc	107
SecurityExchange	207
SecurityID	48
SecurityReqID	320
SecurityRequestType	321
SecurityResponseID	322
SecurityResponseType	323
SecurityStatusReqID	324
SecuritySubType	10762
SecurityTradingStatus	326
SecurityType	167
SenderCompID	49
SenderSubID	50
SendingTime	52
SessionRejectReason	373
SettlType	63
Side	54
StagedOrderMsgTag	16106
StagedOrderOwner	16110
StagedOrderStatus	16109
StopPx	99
StrikePrice	202
SubExchangeGateway	18204

Table 10. FIX tags by name(Continued)

Tag Name	Tag #
SubscriptionRequestType	263
Symbol	55
TargetCompID	56
TargetSubID	57
TestReqID	112
Text	58
TickSize	16452
TimeInForce	59
TotalNumOrders	16728
TotalNumPosReports	16727
TotalNumSecurities	393
TotalVolumeTraded	387
TradeCondition	277
TradeDate	75
TradesInFlow	16464
TransactTime	60
TrdType	10828
TTAccountType	18205
TTOOrderTagData	16105
TTSecurityAlias	18207
TTSessionID	16018
TTStrategyEngine	18208
TTUsername	10553
TTUserTagData	16104
UnderlyingContractTerm	18212
UnderlyingCurrency	318
UnderlyingMaturityDay	314
UnderlyingMaturityMonthYear	313
UnderlyingOptAttribute	317
UnderlyingPutOrCall	315
UnderlyingSecurityAltID	10456
UnderlyingSecurityExchange	308
UnderlyingSecurityID	309
UnderlyingSecurityType	310
UnderlyingStrikePrice	316
UnderlyingSymbol	311
UnsolicitedIndicator	325

Table 10. FIX tags by name(Continued)

FIX tags by number

About this list This section provides a list of FIX tags sorted by tag number.

Tags listed by number

Tag #	Tag Name
1	Account
6	AvgPx
7	BeginSeqNo
8	BeginString
9	BodyLength
10	Checksum
11	ClOrdID
14	CumQty
15	Currency
16	EndSeqNo
17	ExecID
18	ExecInst
19	ExecRefID
20	ExecTransType
21	HandlInst
31	LastPx
32	LastShares
34	MsgSeqNum
35	MsgType
36	NewSeqNo
37	OrderID
38	OrderQty
39	OrdStatus
40	OrdType
41	OrigClOrdID
43	PossDupFlag
44	Price
45	RefSeqNum
47	Rule80A
48	SecurityID
49	SenderCompID
50	SenderSubID
52	SendingTime
54	Side
55	Symbol

Table 11. FIX tags by number

Tag #	Tag Name
56	TargetCompID
57	TargetSubID
58	Text
59	TimeInForce
60	TransactTime
63	SettlType
75	TradeDate
76	ExecBroker
77	OpenClose
96	RawData
97	PossResendFlag
98	EncryptMethod
99	StopPx
102	CxlRejReason
103	OrdRejReason
107	SecurityDesc
108	HeartBtInt
110	MinQty
112	TestReqID
116	OnBehalfOfSubID
122	OrigSendingTime
123	GapFillFlag
129	DeliverToSubID
136	NoMiscFees
137	MiscFeeAmt
141	ResetSeqNumFlag
146	NoRelatedSym
150	ExecType
151	LeavesQty
167	SecurityType
198	SecondaryOrderID
200	MaturityMonthYear
201	PutOrCall
202	StrikePrice
204	CustomerOrFirm
205	MaturityDay
206	OptAttribute
207	SecurityExchange
210	MaxShow
262	MDReqID

Table 11. FIX tags by number(Continued)

Tag #	Tag Name
263	SubscriptionRequestType
264	MarketDepth
265	MDUpdateType
266	AggregatedBook
267	NoMDEntryTypes
268	NoMDEntries
269	MDEntryType
270	MDEntryPx
271	MDEntrySize
277	TradeCondition
279	MDUpdateAction
288	MDEntryBuyer
289	MDEntrySeller
290	MDEntryPositionNo
308	UnderlyingSecurityExchange
309	UnderlyingSecurityID
310	UnderlyingSecurityType
311	UnderlyingSymbol
313	UnderlyingMaturityMonthYear
314	UnderlyingMaturityDay
315	UnderlyingPutOrCall
316	UnderlyingStrikePrice
317	UnderlyingOptAttribute
318	UnderlyingCurrency
319	RatioQty
320	SecurityReqID
321	SecurityRequestType
322	SecurityResponseID
323	SecurityResponseType
324	SecurityStatusReqID
325	UnsolicitedIndicator
326	SecurityTradingStatus
346	NumberOfOrders
371	RefTagID
372	RefMsgType
373	SessionRejectReason
375	ContraBroker
378	ExecRestatementReason
379	BusinessRejectRefID
380	BusinessRejectReason

Table 11. FIX tags by number(Continued)

Tag #	Tag Name
387	TotalVolumeTraded
393	TotalNumSecurities
432	ExpireDate
434	CxlRejResponseTo
440	ClearingAccount
442	MultiLegReportingType
864	NoEvents
865	EventType
866	EventDate
6038	OrderEnteredTime
10442	ExpectedFillType
10455	SecurityAltID
10456	UnderlyingSecurityAltID
10527	SecondaryExecID
10541	FixingDate
10553	TTUsername
10566	LegPrice
10762	SecuritySubType
10828	TrdType
11028	ManualOrderIndicator
16017	OrigExecID
16018	TTSessionID
16102	FFT2
16103	FFT3
16104	TTUserTagData
16105	TTOrderData
16106	StagedOrderMsgTag
16109	StagedOrderStatus
16110	StagedOrderOwner
16111	RoutingLevel
16112	NoLinks
16113	LinkID
16114	LinkType
16142	OrderOriginationID
16210	RealizedPnDL
16309	ExchContractSymbol
16451	PriceDisplayType
16452	TickSize
16454	PointValue
16456	NumTickTblEntries

Table 11. FIX tags by number(Continued)

Tag #	Tag Name
16457	NumTicks
16458	MaxPrice
16460	DeliveryUnit
16461	LotSize
16463	Blocks
16464	TradesInFlow
16480	PassiveAggressive
16481	AutoAggressive
16486	MDEntryState
16487	MDEntrySizeType
16488	AggressorSide
16489	ExactNumOrdersIndicator
16552	ExchTickSize
16554	ExchPointValue
16624	LegSide
16710	PosReqId
16721	PosMaintRptId
16724	PosReqType
16727	TotalNumPosReports
16728	TotalNumOrders
16899	MarginExcess
17000	RequestTickTable
18000	ForceLogout
18200	GatewayStatusReqId
18201	NoGatewayStatus
18202	GatewayStatus
18203	ExchangeGateway
18204	SubExchangeGateway
18205	TTAccountType
18206	NoGateways
18207	TTSecurityAlias
18208	TTStrategyEngine
18210	PriceFeedStatus
18211	ContractTerm
18212	UnderlyingContractTerm
18214	IncludeNumberOfOrders
18216	ExchCred

Table 11. FIX tags by number(Continued)

Overview

This appendix describes how you can calculate a trader's total P&L from the realized P&L provided by TT FIX Adapter in Tag 16210 (**RealizedP&L**) and the unrealized P&L you calculate from a theoretical exit price. For any contract, you can use the following formula to determine total P&L:

$$P\&L_{Total} = P\&L_{Realized} + P\&L_{Unrealized}$$

You can choose from many fill-matching methodologies when calculating realized and unrealized P&L, such as FIFO, LIFO, and Averaging, and so on. TT FIX Adapter uses the Averaging methodology because it provides a distinct speed advantage over the other methods. (The Averaging methodology is mathematical whereas the LIFO and FIFO methodologies are algorithmic.)

Note: Each methodology might derive different values when calculating values for intermediate variables, but all of them arrive at the same result for total P&L.

Illustrating total P&L calculations

To help you understand how you can calculate total P&L, this appendix uses a set of hypothetical fills to establish a baseline position and realized P&L. Then, using these fills, it shows you how to calculate both unrealized and total P&L.

These P&L values represent the status after the initial fill download. Rather than using all of the fills to recalculate P&L when additional fills arrive, you can streamline performance by saving the initial state and basing future calculations on only the new fills. To demonstrate, this example walks you through calculations in the following scenarios:

- [Scenario 1: receiving new fills that increase your position](#)
- [Scenario 2: receiving new fills that partially decrease your position](#)
- [Scenario 3: receiving fills that flatten your position](#)
- [Scenario 4: receiving fills that reverse your position](#)

Initial fill download

Assume the initial fill download includes the following fills (all prices in points):

- Buy 12 @ 100
- Buy 17 @ 99
- Sell 9 @ 101
- Sell 4 @ 105
- Buy 3 @ 103

TT FIX Adapter uses the following formula to calculate the average price of the buys and sells.

$$\frac{\sum_{i=1}^{fills} ([Qty]_i \cdot [Price]_i)}{\sum_{i=1}^{fills} [Qty]_i}$$

Initial P&L calculations

From these fills, TT FIX Adapter calculates the following base values:

- Total Buy Quantity = $12 + 17 + 3 = 32$
- Average Buy Price = $((12 * 100) + (17 * 99) + (3 * 103)) / 32 = 99.75$ (points)
- Total Sell Quantity = $9 + 4 = 13$
- Average Sell Price = $((9 * 101) + (4 * 105)) / 13 = 102.230769$ (points)

To determine the realized P&L, TT FIX Adapter matches thirteen Buys with thirteen Sells using the Averaging technique, as follows:

- $P\&L_{Realized}$ (points) = (Sell Price - Buy Price) * Qty = $(102.230769 - 99.75) * 13 = 32.249997$
- $P\&L_{Realized}$ (contract currency) = $P\&L_{Realized}$ (points) * Contract Point Value

which results in the following starting state after the initial fill download:

- Position = +19
- Average Open Price = 99.75

With the Average Open Price for the initial fills, the FIX client can calculate the unrealized P&L for the initial position. To do so, you must use some Theoretical Exit Price to calculate the unrealized P&L. This example, and all of the scenarios, assumes a Theoretical Exit Price of 99 (points), which results in the following calculations:

- $P\&L_{Unrealized}$ (points) = (Theoretical Exit Price - Average Open Price) * Position = $(99 - 99.75) * (+19) = -14.25$
- $P\&L_{Unrealized}$ (contract currency) = $P\&L_{Unrealized}$ (points) * Contract Point Value

With these values, you can calculate the total P&L after the initial fill download as follows:

- $P\&L_{Total}$ (points) = $P\&L_{Realized}$ (points) + $P\&L_{Unrealized}$ (points) = $32.249997 + (-14.25) = 17.999997$
- $P\&L_{Total}$ (contract currency) = $P\&L_{Total}$ (points) * Contract Point Value

Scenario 1: receiving new fills that increase your position

After calculating the initial position and P&L, suppose you receive a fill that increases your position (adding Buys to a positive position or Sells to a negative one). In this example, you receive a fill for a Buy order, as follows:

Start state	Position = +19 Average Open Price = 99.75
New Fill	Buy 10 @ 100

In this scenario, the fills do not affect the realized P&L, but they do affect the unrealized and total P&L, as follows:

- Position = $19 + 10 = +29$
- Average Open Price = $((19 * 99.75) + (10 * 100)) / (19 + 10) = 99.83620$

Scenario 2: receiving new fills that partially decrease your position

Instead of increasing a position, suppose the new fill partially decreases the position, still leaving an open position. In this example, you receive a fill for a Sell order, as follows:

Start state	P&L _{Realized} (points) = 32.249997 Position = +19 Average Open Price = 99.75
New Fill	Sell 12 @ 101

In this scenario, the reduced open position does affect the realized P&L, as well as the unrealized and total P&L. The following calculations show how TT FIX Adapter derives the new realized P&L and how you can calculate the updated unrealized and total P&L (with Theoretical Exit Price = 99):

- $P\&L_{Realized}$ (points) = $P\&L_{Realized} + ((\text{Sell Price} - \text{Buy Price}) * \text{Qty})$
= $32.249997 + ((101 - 99.75) * 12) = 47.249997$
- Position = $19 - 12 = 7$
- Average Open Price = 99.75 (does not change)
- $P\&L_{Unrealized}$ (points) = $(\text{Theoretical Exit Price} - \text{Average Open Price}) * \text{Position}$
= $(99 - 99.75) * 7 = -5.25$
- $P\&L_{Total}$ (points) = $(P\&L_{Realized} \text{ (points)}) + (P\&L_{Unrealized} \text{ (points)}) =$
 $47.249997 + (-5.25) = 41.999997$

Scenario 3: receiving fills that flatten your position

When a fill flattens a position, the average price becomes unavailable, as does the unrealized P&L. In this example, you receive a fill for a Sell order that matches the current position, as follows:

Start state	P&L _{Realized} (points) = 32.249997 Position = +19 Average Open Price = 99.75
New Fill	Sell 19 @ 101

In this scenario, flattening a position causes the realized and total P&L to derive the same value. The following calculations show how TT FIX Adapter derives the new realized P&L and how you can calculate the updated unrealized and total P&L (with Theoretical Exit Price = 99):

- $P\&L_{Realized}$ (points) = $P\&L_{Realized} + ((\text{Sell Price} - \text{Buy Price}) * \text{Qty})$
= $32.249997 + ((101 - 99.75) * 19) = 55.999997$
- Position = $19 - 19 = 0$
- Average Open Price = none (as position = 0)
- $P\&L_{Unrealized}$ (points) = $(\text{Theoretical Exit Price} - \text{Average Open Price}) * \text{Position}$
= none (as position = 0)
- $P\&L_{Total}$ (points) = $(P\&L_{Realized} \text{ (points)}) + (P\&L_{Unrealized} \text{ (points)}) =$
 $55.999997 + 0$
= 55.999997

Scenario 4: receiving fills that reverse your position

Finally, you can also receive fills that reverse a position. In this example, you receive a fill for a Sell order than exceeds the current position, as follows:

Start state	$P\&L_{Realized}$ (points) = 32.249997 Position = +19 Average Open Price = 99.75
New Fill	Sell 22 @ 101

In this scenario, reversing the position affects the realized P&L, as well as the unrealized and total P&L. The following calculations show how TT FIX Adapter derives the new realized P&L and how you can calculate the updated unrealized and total P&L (with Theoretical Exit Price = 99):

- $P\&L_{Realized}$ (points) = $P\&L_{Realized}$ + ((Sell Price - Buy Price) * Qty)
 = $32.249997 + ((101 - 99.75) * 19) = 55.999997$
- Position = $19 - 22 = -3$
- Average Open Price = 101
- $P\&L_{Unrealized}$ (points) = (Theoretical Exit Price - Average Open Price) * Position
 = $(99 - 101) * (-3) = 6$
- $P\&L_{Total}$ (points) = ($P\&L_{Realized}$ (points)) + ($P\&L_{Unrealized}$ (points)) =
 $55.999997 + 6 = 61.999997$

Appendix overview

This appendix provides information that applies only for specific TT Gateways.

In this chapter

Section	Page
BrokerTec Notes	230
CBOE Notes	232
Eurex Notes	232
NYSE Liffe Notes	233
ICE Notes	234
TFX Notes	235
QM Gateway Notes	236

BrokerTec Notes

Quoting conventions	<p>BrokerTec quotes some products (such as WI and JGB bonds) in yield. When displaying these quotes, X_TRADER also shows them in yield. TT FIX Adapter, however, expects and delivers prices for these products as 100-yield.</p> <p>For example, if BrokerTec sends a quote for a 5% yield, TT FIX Adapter sends the FIX client the price as 95 in Tag 270 (MDEntryPx). Consequently, when submitting orders for WI products, the FIX client must submit 95 in Tag 44 (Price).</p>
Changing or rejecting orders	<p>BrokerTec does not allow you to change, cancel, replace, or delete orders that reside at the inside market for a specified period of time once entered (usually about 2 seconds, but it can change). If you try to modify an order during this time period, BrokerTec sends a request reject with a special flag that indicates this particular situation. In this case, TT FIX Adapter sends an Order Cancel Reject (9) message to the FIX client with Tag 102 (CxlRejReason) set to 4 (Too Early To Cancel).</p>
Fee information in fills during workup	<p>For BrokerTec, fills that occur during the workup phase do not include the fee information. Instead, this information is included in an updated fill that is sent after the workup phase is over. Consequently, TT FIX Adapter does not include Tag 136 (NoMiscFees) for fills during the workup phase, but does include it include it when BrokerTec provides the updated fill after the workup phase. Note that you must have the Send Fill Updates option enabled in FACT to receive updated fill notifications.</p>
WI and OTR Bond Support	<p>The BrokerTec Gateway supports WI (When Issued) and OTR (On-The-Run) bonds. For US Treasuries, when a new WI or OTR bond becomes available on the BrokerTec Gateway, the Tag 48 (SecurityID) value will be the same as that of the previous WI or OTR bond, but the Tag 10455 (SecurityAltID), Tag 205 (MaturityDay), and Tag 200 (MaturityMonthYear) values will change. See the <i>BrokerTec Gateway System Administration Manual</i> for more information regarding the representation of BrokerTec instruments.</p>

CBOE Notes

TT CBOE Gateway support for change orders

The TT CBOE Gateway does not currently support change order requests. Consequently, TT FIX Adapter rejects **Order Cancel Replace Request (G)** messages sent from FIX clients for TT CBOE Gateway contracts.

Modifying orders

If a FIX client wants to modify an order on the TT CBOE Gateway, it must first submit an **Order Cancel Request (F)** message to cancel the working order and then submit a **New Order Single (D)** message with the updated order information.

Eurex Notes

About over-the-counter trades	<p>Over-the-counter (OTC) trades occur directly between two parties where the exchange acts as a clearing mechanism for the transaction. Typically, OTC trades fall into one of the following types:</p> <ul style="list-style-type: none">• Type 1 trades, which include block, vola, and similar trades, where the trade comprises a cash leg and a standard exchange-listed futures or options leg• Type 2 trades, like FLEX options, where the trade comprises a cash leg and a non-standard contract leg• Type 3 trades, including EFP, EFS, AA, and similar trades, where the trade comprises a non-standard contract leg (bond, swap, commodity) and a standard exchange-listed futures or options leg.
Eurex OTC support	<p>The Eurex Gateway supports only Type 1 trades.</p>
How TT FIX Adapter handles Eurex OTC trades	<p>While TT FIX Adapter does not support order routing for OTC trades, traders can submit OTC orders through X_TRADER®. TT FIX Adapter does, however, process the corresponding fills. For an OTC trade submitted through X_TRADER®, TT FIX Adapter sends an Execution Report (8) message when the exchange-listed futures or options leg of the trade fills.</p>

NYSE_Liffe Notes

About over-the-counter trades

Over-the-counter (OTC) trades occur directly between two parties where the exchange acts as a clearing mechanism for the transaction. Typically, OTC trades fall into one of the following types:

- Type 1 trades, which include block, vola, prof, and similar trades, where the trade comprises a cash leg and a standard exchange-listed futures or options leg
- Type 2 trades, like FLEX options, where the trade comprises a cash leg and a non-standard contract leg
- Type 3 trades, including EFP, EFS, AA, and similar trades, where the trade comprises a non-standard contract leg (bond, swap, commodity) and a standard exchange-listed futures or options leg

OTC support

The TT NYSE_Liffe Gateway supports all three types of OTC trades. Except for “prof” trades, both TT Gateways also require that one party enter both sides of a transaction.

How TT FIX Adapter handles OTC trades

While TT FIX Adapter does not support entering OTC trades, traders can submit OTC orders through X_TRADER[®]. TT FIX Adapter does, however, process the corresponding fills and delivers them to FIX clients. For an OTC trade submitted through X_TRADER[®], TT FIX Adapter responds differently for each trade type, as described in Table 12.

For OTC trade type...	TT FIX Adapter...
Type 1	For all except prof trades: Sends to the order sender two Execution Report (8) messages for the exchange-traded leg fill, one for the buyer and one for the seller. Clients should use Tag 1 (Account) to determine which fills are theirs. For prof trades: Sends the order sender a single Execution Report (8) message for the exchange-traded leg fill.
Type 2	Sends nothing.
Type 3	Sends to the order sender two Execution Report (8) messages for the exchange-traded leg fill, one for the buyer and one for the seller. Clients should use Tag 1 (Account) to determine which fills are theirs.

Table 12. TT FIX Adapter responses for LIFFE OTC trades

ICE Notes

Uniquely distinguishing ICE contracts

The ICE Exchange might list a futures and energy contract with the same maturity date, but with different delivery terms. When receiving messages from TT FIX Adapter for these types of contracts, FIX clients can use either Tag 10455 (**SecurityAltID**) or Tag 18211 (**ContractTerm**) to distinguish between them.

Example: The following shows how you can use these tags to distinguish between three different ICE Coal ARA contracts with an Oct 07 maturity date:

```
55=ICE Coal ARA#167=FUT+207=ICE_IPE#200=200710#10455=Q4 07#18211=Q
55=ICE Coal ARA#167=FUT+207=ICE_IPE#200=200710#10455=Winter07#18211=S
55=ICE Coal ARA#167=FUT+207=ICE_IPE#200=200710#10455=Oct07
```

For ICE contracts with monthly delivery terms, TT FIX Adapter does not send Tag 18211 (**ContractTerm**) in the Instrument Component Block.

Routing orders using the "by name" method

When routing orders to the TT FIX Adapter using the "by name" method, TT FIX Adapter supports only contracts that have monthly, seasonal, quarterly, or yearly delivery terms. For contracts with monthly delivery terms, FIX clients should not include Tag 18211 (**ContractTerm**). For example, to route an order for the Oct07 ICE Coal ARA future, you would specify:

```
55=ICE Coal ARA#167=FUT+207=ICE_IPE#200=200710
```

To route an order for the Winter07 ICE Coal ARA future, you would specify:

```
55=ICE Coal ARA#167=FUT+207=ICE_IPE#200=200710#18211=S
```

TT FIX Adapter does not support routing orders for ICE contracts with daily, weekly, and variable delivery terms. If traders place such orders from other trading applications, such as X_TRADER, TT FIX Adapter does send their corresponding order acknowledgements and fills to FIX clients.

Note: The ICE simulation environment might list duplicate contracts. You should exercise caution when testing your application to ensure that it routes orders for the correct contracts.

TFX Notes

About over-the-counter trades	<p>Over-the-counter (OTC) trades occur directly between two parties where the exchange acts as a clearing mechanism for the transaction. Typically, OTC trades fall into one of the following types:</p> <ul style="list-style-type: none">• Type 1 trades, which include block, vola, and similar trades, where the trade comprises a cash leg and a standard exchange-listed futures or options leg• Type 2 trades, like FLEX options, where the trade comprises a cash leg and a non-standard contract leg• Type 3 trades, including EFP, EFS, AA, and similar trades, where the trade comprises a non-standard contract leg (bond, swap, commodity) and a standard exchange-listed futures or options leg.
TFX OTC support	<p>The TT TFX Gateway supports only Type 1 trades.</p>
How TT FIX Adapter handles Eurex OTC trades	<p>While TT FIX Adapter does not support order routing for OTC trades, traders can submit OTC orders through X_TRADER®. TT FIX Adapter does, however, process the corresponding fills. For an OTC trade submitted through X_TRADER®, TT FIX Adapter sends an Execution Report (8) message when the exchange-listed futures or options leg of the trade fills.</p>

OM Gateway Notes

Supported OM Gateways

TT FIX Adapter supports the following OM platform Gateways:

- BrokerTec
- OSE
- SGX
- TOCOM

How TT FIX Adapter handles corrected fills

The various OM Gateways often send two fill records for each fill.

- 1 When an order receives a fill, the OM Gateways send the initial fill message to the TT FIX Adapter, which immediately sends FIX clients an **Execution Report (8)** message.
- 2 If the initial fill omits some of the data used during the clearing process that was not immediately available, the OM Gateways send a second fill message to the TT FIX Adapter. If the **Send Fill Updates** option is selected in FACT, the TT FIX Adapter then sends the FIX clients an **Execution Report (8)** message representing the updated fill.

The second **Execution Report (8)** message can include any of the following tags:

- Tag 40 (**OrdType**)
- Tag 44 (**Price**)
- Tag 59 (**TimeInForce**)
- Tag 60 (**TransactTime**)
- Tag 75 (**TradeDate**)
- Tag 77 (**OpenClose**)
- Tag 99 (**StopPx**)
- Tag 440 (**ClearingAccount**)
- Tag 10527 (**SecondaryExecID**)
- Tag 16102 (**FFT2**)

LME Gateway Notes

Tag 48 and Tag 10455 Values for LME Contracts

LME contracts will roll through several different Tag 48 (**SecurityID**) and Tag 10455 (**SecurityAltID**) values throughout their lifetimes. This is because the LME Gateway will represent these contracts with *aliases* (3M, TOM, and C) on certain days, for one day per alias, based on the number of days away from that contract's prompt date. See the *LME Gateway System Administration Manual* for more information regarding the LME Gateway's representation of LME contracts throughout their lifetime.

Note: While the Tag 48 (**SecurityID**) and Tag 10455 (**SecurityAltID**) values will vary, the values for the following will be constant for a given LME contract throughout its lifetime: Tag 18203 (**ExchangeGateway**), Tag 200 (**MaturityMonthYear**), Tag 205 (**MaturityDay**), Tag 167 (**SecurityType**), Tag 207 (**SecurityExchange**), and Tag 55 (**Symbol**).

Tag 200 and Tag 205 Values for LME instruments

For LME instruments, Tag 200 (**MaturityMonthYear**) and Tag 205 (**MaturityDay**) are populated with the contract's prompt date. As such, the Tag 200 (**MaturityMonthYear**) and Tag 205 (**MaturityDay**) values of LME's *aliased* rolling prompt contracts (3M, TOM, C) change every day, even though they will reuse the same Tag 48 (**SecurityID**) and Tag 10455 (**SecurityAltID**) values.

Index

Symbols

117

A

Account tag 49, 51, 126, 136, 139, 141
AggregatedBook tag 79
AggressorSide tag 86, 92
AutoAggressive tag 114, 124, 133, 144
AvgPx tag 115

B

BeginSeqNo tag 70
BeginString tag 31, 32
Blocks tag 100
BodyLength tag 31, 33
BusinessRejectReason tag 157
BusinessRejectRefID tag 157

C

Checksum tag 61
ClearingAccount tag 49, 51
ClOrdID tag 121, 126, 136
Configuration file
 TargetSubID parameter 32
ContraBroker tag 114, 144
ContractTerm tag 40, 42
CumQty tag 111
Currency tag 43
CustomerOrFirm tag 50, 52
CxlRejReason tag 127
CxlRejResponseTo tag 127

D

DeliverToSubID tag 34
DeliveryUnit tag 100

E

EncryptMethod tag 65
EndSeqNo tag 70
EventDate tag 100
EventType tag 100
ExactNumOrdersIndicator tag 87, 94
ExchangeGateway tag 100, 115, 125, 133, 147, 160
ExchContractSymbol tag 99
ExchCred tag 148
ExchPointValue tag 99
ExchTickSize tag 99

ExecBroker tag 114, 144
ExecID tag 107, 146
ExecInst tag 121, 130
ExecRefID tag 107, 146
ExecRestatementReason tag 114
ExecTransType tag 107, 146
ExecType tag 109
ExpectedFillType tag 98
ExpireDate tag 114, 124, 131, 144

F

FFT2 tag 49, 51
FFT3 tag 49, 51
FIX message
 structure 29
FIX protocol version 31, 32
FixingDate tag 114, 144
ForceLogout tag 67

G

GapFillFlag tag 71
Gateways
 supported 14
GatewayStatus tag 159
GatewayStatusReqId tag 158, 159

H

HandInst tag 108, 121, 130, 147
HeartBtInt tag 65

I

IncludeNumberOfOrders tag 80
Instrument
 specify by alternate ID 38
 specify by ID 38
 specify by name 39
Instrument component block
 from FIX Adapter 41
 from FIX client 37
 underlying instruments 44
Instruments
 underlying 44

L

LastPx tag 115, 145
LastShares tag 110, 145
LeavesQty tag 110
LegPrice tag 45, 48
LegSide tag 45, 48

Index

LinkID tag 117, 148
LinkType tag 117, 148
LotSize tag 99

M

ManualOrderIndicator tag 116, 124, 131, 134
MarginExcess tag 99
MarketDepth tag 79
MaturityDay tag 39, 40, 41, 42
MaturityMonthYear tag 39, 41
MaxPrice tag 99
MaxShow tag 112, 122, 131, 142
MDEntryBuyer tag 93
MDEntryPositionNo tag 86, 92
MDEntryPx tag 84, 90
MDEntrySeller tag 93
MDEntrySize tag 85, 91
MDEntrySizeType tag 85, 91
MDEntryState tag 87, 93
MDEntryType tag 80, 84, 90
MDReqID tag 79, 82, 83, 89
MDUpdateAction tag 89
MDUpdateType tag 79
Message
 header 31
 structure 29
Message body
 common components 36
 instrument component block 37, 41
message body 36
Message header
 tags 31, 32
Message structure
 message body 36
 message header 31
 parts of 30
 trader component block 49
Messages
 common components 36
MinQty tag 112, 122, 130, 142
MiscFeeAmt tag 117, 148
MsgSeqNum tag 31, 33
MsgType tag 31, 33
MultiLegReportingType tag 110, 147

N

NewSeqNo tag 71
NoEvents tag 100
NoGateways tag 100
NoGatewayStatus tag 159
NoLinks tag 117, 148
NoMDEntries tag 84, 89
NoMDEntryTypes tag 80
NoMiscFees tag 117, 148
NoRelatedSym tag 44, 46, 80, 146
NumberOfOrders tag 87, 93
NumTicks tag 99

NumTickTblEntries tag 99

O

OnBehalfOfSubID tag 32, 34
OpenClose tag 113, 124, 131, 143
OptAttribute tag 40, 42
OrderEnteredTime tag 115
OrderID tag 108, 126, 129, 134, 136, 146
OrderOriginationID tag 116, 124, 131, 134
OrderQty tag 112, 122, 130, 142
OrderStatus tag 110, 127
OrdRejReason tag 109
OrdType tag 113, 123, 132, 143
OrigClOrdID tag 109, 126, 129, 134
OrigExecID tag 108, 146
OrigSendingTime tag 32, 34

P

PassiveAggressive tag 114, 124, 132, 144
PointValue tag 99
PosMaintRptId tag 145
PosReqId tag 139
PosReqType tag 139, 145
PossDupFlag tag 31, 34
PossResendFlag tag 34
Price tag 111, 122, 130, 142
PriceDisplayType tag 98
PriceFeedStatus tag 83, 89
PutOrCall tag 40, 42

R

RatioQty tag 46, 48
RawData tag 65
RealizedPnL tag 145
RefMsgType tag 72, 157
RefSeqNum tag 72, 157
RefTagID tag 72
Regulations, CME and CBOT 35
RequestTickTable tag 96
ResetSeqNumFlag tag 65
RoutingLevel tag 116, 125, 133
Rule80A tag 50, 52

S

SecondaryExecID tag 115, 147
SecondaryOrderID tag 108, 126
Security
 specify by alternate ID 38
 specify by ID 38
 specify by name 39
SecurityAltID tag 38, 43
SecurityDesc tag 100
SecurityExchange tag 38, 39, 41, 97, 159
SecurityID tag 38, 43, 97

Index

SecurityReqID tag	96, 98		
SecurityRequestType tag	96		
SecurityResponseID tag	98		
SecurityResponseType tag	98		
SecurityStatusReqID tag	102, 103		
SecuritySubType tag	44, 46		
SecurityTradingStatus tag	103		
SecurityType tag	39, 40, 41, 42, 96		
SenderCompID tag	31, 33		
SenderSubID tag	34		
SendertSubID tag	32		
SendingTime tag	32, 34		
SessionRejectReason tag	72		
SettlType tag	40, 42		
Side tag	122, 131		
StagedOrderMsg tag	116, 125, 133		
StagedOrderOwner tag	116		
StagedOrderStatus tag	116		
StopPrice tag	112, 122, 130, 142		
StrikePrice tag	40, 42		
SubExchangeGateway tag	160		
SubscriptionRequestType tag	79, 102, 139, 158		
Supported TT Gateways	14		
Symbol tag	38, 39, 41, 96		
			147, 160
		ExchContractSymbol(16309)	99
		ExchCred (18216)	117, 148
		ExchPointValue(16554)	99
		ExchTickSize(16552)	99
		ExecBroker (76)	114, 144
		ExecID (17)	107, 146
		ExecInst (18)	121, 130
		ExecRefID (19)	107, 146
		ExecRestatementReason (378)	114
		ExecTransType (20)	107, 146
		ExecType (150)	109
		ExpectedFillType (10442)	98
		ExpireDate (432)	114, 124, 131, 144
		FFT2 (16102)	49, 51
		FFT3 (16103)	49, 51
		FixingDate(10541)	114, 144
		ForceLogout (18000)	67
		GapFillFlag (123)	71
		GatewayStatus (18202)	159
		GatewayStatusReqId (18200)	158, 159
		HandInst (21)	108, 121, 130, 147
		HeartBtInt (108)	65
		IncludeNumberOfOrders (18214)	80
		LastPx (31)	115, 145
		LastShares (32)	110, 145
		LeavesQty (151)	110
		LegPrice (10566)	45, 48
		LegSide (16624)	45, 48
		LinkID (16113)	117, 148
		LinkType (16114)	117, 148
		LotSize(16461)	99
		ManualOrderIndicator(11028)	116, 124, 131, 134
		MarginExcess (16899)	99
		MarketDepth (264)	79
		MaturityDay (205)	39, 40, 41, 42
		MaturityMonthYear (200)	39, 41
		MaxPrice(16458)	99
		MaxShow (210)	112, 122, 131, 142
		MDEntryBuyer (288)	93
		MDEntryPositionNo (290)	86, 92
		MDEntryPx (270)	84, 90
		MDEntrySeller (289)	93
		MDEntrySize (271)	85, 91
		MDEntrySizeType(16487)	85
		MDEntrySizeTypee(16487)	91
		MDEntryState(16486)	87, 93
		MDEntryType (269)	80, 84, 90
		MDReqID (262)	79, 82, 83, 89
		MDUpdateAction (279)	89
		MDUpdateType (265)	79
		MinQty (110)	112, 122, 130, 142
		MiscFeeAmt (136)	117, 148
		MsgSeqNum (34)	31, 33
		MsgType (35)	31, 33
		MultiLegReportingType (442)	110, 147
		NewSeqNo (36)	71
		NoEvents (864)	100
		NoGateways (18206)	100
		NoGatewayStatus (18201)	159
		NoLinks (16112)	117, 148
T			
Tags			
message header	31, 32		
Tags (by name)			
Account (1)	49, 51, 126, 136, 139, 141		
AggregatedBook (266)	79		
AggressorSide(16488)	86, 92		
AutoAggressive(16481)	114, 124, 133, 144		
AvgPx (6)	115		
BeginSeqNo (7)	70		
BeginString (8)	31, 32		
Blocks(16463)	100		
BodyLength (9)	31, 33		
BusinessRejectReason (380)	157		
BusinessRejectRefID (379)	157		
Checksum (10)	61		
ClearingAccount (440)	49, 51		
ClOrdID (11)	121, 126, 136		
ContraBroker (375)	114, 144		
ContractTerm (18211)	40, 42		
CumQty (14)	111		
Currency (15)	43		
CustomerOrFirm (204)	50, 52		
CxlRejReason (102)	127		
CxlRejResponseTo (434)	127		
DeliverToSubID (129)	34		
DeliveryUnit(16460)	100		
EncryptMethod (98)	65		
EndSeqNo (16)	70		
EventDate (866)	100		
EventType (865)	100		
ExactNumOrdersIndicator (10828)	117, 148		
ExactNumOrdersIndicator (16489)	87, 94		
ExchangeGateway (18203)	100, 115, 125, 133,		

NoMDEntries (268)	84, 89	SessionRejectReason (373)	72
NoMDEntryTypes (267)	80	SettlType (63)	40, 42
NoMiscFees (136)	117, 148	Side (54)	122, 131
NoRelatedSym (146)	44, 46, 80, 146	StagedOrderMsg (16106)	116, 125, 133
NumberOfOrders (346)	87, 93	StagedOrderOwner (16110)	116
NumTicks(16457)	99	StagedOrderStatus (16109)	116
NumTickTblEntries(16456)	99	StopPrice (99)	112, 122, 130, 142
OnBehalfOfSubID (116)	32, 34	StrikePrice (202)	40, 42
OpenClose (77)	113, 124, 131, 143	SubExchangeGateway (18204)	160
OptAttribute (206)	40, 42	SubscriptionRequestType (263 79, 102, 139, 158)	
OrderEnteredTime (6038)	115	Symbol (55)	38, 39, 41, 96
OrderID (37)	108, 126, 129, 134, 136, 146	TargetCompID (56)	31, 33
OrderOriginationID(16142)	116, 124, 131, 134	TargetSubID (57)	32, 33, 34
OrderQty (38)	112, 122, 130, 142	TestReqId (112)	69
OrderStatus (39)	110, 127	Text (58)	103, 111, 127, 147, 160
OrdRejReason (103)	109	TickSize(16452)	99
OrdType (40)	113, 123, 132, 143	TimeInForce (59)	114, 124, 131, 144
OrigClOrdID (41)	109, 126, 129, 134	TotalNumOrders (16728)	111
OrigExecId(16017)	108, 146	TotalNumPosReports (16727)	145
OrigSendingTime (122)	32, 34	TotalNumSecurities (393)	98
PassiveAggressive(16480)	114, 124, 132, 144	TotalVolumeTraded (387)	83, 92
PointValue(16454)	99	TradeConnection (277)	87, 93, 115
PosMaintRptId (16721)	145	TradeDate (75)	114, 144
PosReqId (16710)	139	TradesInFlow(16464)	100
PosReqType (16724)	139, 145	TransactTime (60)	115, 121, 127, 129, 147
PossDupFlag (43)	31, 34	TTAccountType (18205)	50, 52
PossResendFlag (97)	34	TTOderTagData (16105)	116, 125, 133, 147
Price (44)	111, 122, 130, 142	TTSecurityAlias (18207)	100
PriceDisplayType(16451)	98	TTSessionID(16018)	115, 147
PriceFeedStatus (18210)	83, 89	TTStrategyEngine (18208)	116, 147
PutOrCall (201)	40, 42	TTUserTagData (16104)	116, 125, 133, 147
RatioQty (319)	46, 48	UnderlyingContractTerm (18212)	45, 47
RawData (96)	65	UnderlyingCurrency (318)	47
RealizedPnDL(16210)	145	UnderlyingMaturityDay (314)	45, 47
RefMsgType (372)	72, 157	UnderlyingMaturityMonthYear (313)	45, 46
RefSeqNum (45)	72, 157	UnderlyingOptAttribute (317)	45, 47
RefTagID (371)	72	UnderlyingPutOrCall (315)	45, 47
RequestTickTable (17000)	96	UnderlyingSecurityAltID (10456)	47
ResetSeqNumFlag (141)	65	UnderlyingSecurityExchange (308)	44, 46
RoutingLevel (16111)	116, 125, 133	UnderlyingSecurityID (309)	47
Rule80A (47)	50, 52	UnderlyingSecurityType (310)	44, 46
SecondaryExecID(10527)	115, 147	UnderlyingStrikePrice (316)	45, 47
SecondaryOrderID (198)	108, 126	UnderlyingSymbol (311)	44, 46
SecuritAltID (10455)	38, 43	UnsolicitedIndicator (325)	159
SecurityAltID (10455)	38, 43	Username (10553)	53, 126
SecurityDesc (107)	100	Tags (by number)	
SecurityExchange (207)	38, 39, 41, 97, 159	1 (Account)	49, 51, 126, 136, 139, 141
SecurityID (48)	38, 43, 97	10 (Checksum)	61
SecurityID (55)	38, 43	102 (CxlRejReason)	127
SecurityReqID (320)	96, 98	103 (OrdRejReason)	109
SecurityRequestType (321)	96	10442 (ExpectedFillType)	98
SecurityResponseID (322)	98	10455 (SecurityAltID)	38, 43
SecurityResponseType (323)	98	10456 (UnderlyingSecurityAltID)	47
SecurityStatusReqID (324)	102, 103	10527 (SecondaryExecID)	115, 147
SecuritySubType (10762)	44, 46	10541 (FixingDate)	114, 144
SecurityTradingStatus (326)	103	10553 (Username)	53, 126
SecurityType (167)	39, 40, 41, 42, 96	10566 (LegPrice)	45, 48
SecurityType (167)	39, 41	107 (CSecurityDesc)	100
SenderCompID (49)	31, 33	10762 (SecuritySubType)	44, 46
SenderSubID (50)	32, 34	108 (HeartBtInt)	65
SendingTime (52)	32, 34	10828 (TrdType)	117, 148

E TT Trading Networks

11 (ClOrdID)	121, 126, 136
110 (MinQty)	112, 122, 130, 142
11028 (ManualOrderIndicator)	116, 124, 131, 134
112 (TestReqId)	69
116 (OnBehalfOfSubID)	32, 34
122 (OrigSendingTime)	32, 34
123 (GapFillFlag)	71
129 (DeliverToSubID)	34
136 (NoMiscFees)	117, 148
137 (MiscFeeAmt)	117, 148
14 (CumQty)	111
141 (ResetSeqNumFlag)	65
146 (NoRelatedSym)	44, 46, 80, 146
15 (Currency)	43
150 (ExecType)	109
151 (LeavesQty)	110
16 (EndSeqNo)	70
16017 (OrigExecId)	108, 146
16018 (TTSessionID)	115, 147
16102 (FFT2)	49, 51
16103 (FFT3)	49, 51
16104 (TTUserTagData)	116, 125, 133, 147
16105 (TTOrderTagData)	116, 125, 133, 147
16106 (StagedOrderMsg)	116, 125, 133
16109 (StagedOrderStatus)	116
16110 (StagedOrderOwner)	116
16111 (RoutingLevel)	116, 125, 133
16112 (NoLinks)	117, 148
16113 (LinkID)	117, 148
16114 (LinkType)	117, 148
16142 (OrderOriginationID)	116, 124, 131, 134
16210 (RealizedPnDL)	145
16309 (ExchContractSymbol)	99
16451 (PriceDisplayType)	98
16452 (TickSize)	99
16454 (PointValue)	99
16456 (NumTickTblEntries)	99
16457 (NumTicks)	99
16458 (MaxPrice)	99
16460 (DeliveryUnit)	100
16461 (LotSize)	99
16463 (Blocks)	100
16464 (TradesInFlow)	100
16480 (PassiveAggressive)	114, 124, 132, 144
16481 (AutoAggressive)	114, 124, 133, 144
16486 (MDEntryState)	87, 93
16487 (MDEntrySizeType)	85, 91
16488 (AggressorSide)	86, 92
16489 (ExactNumOrdersIndicator)	87, 94
16552 (ExchTickSize)	99
16554 (ExchPointValue)	99
16624 (LegSide)	45, 48
167 (SecurityType)	39, 40, 41, 42, 96
16710 (PosReqId)	139
16721 (PosMaintRptId)	145
16724 (PosReqType)	139, 145
16727 (TotalNumPosReports)	145
16728 (TotalNumOrders)	111
16899 (MarginExcess)	99
17 (ExecID)	107, 146
17000 (RequestTickTable)	96

18 (ExecInst)	121, 130
18000 (ForceLogout)	67
18200 (GatewayStatusReqId)	158, 159
18201 (NoGatewayStatus)	159
18202 (GatewayStatus)	159
18203 (ExchangeGateway)	100, 115, 125, 133, 147, 160
18204 (SubExchangeGateway)	160
18205 (TTAccountType)	50, 52
18206 (NoGateways)	100
18207 (TTSecurityAlias)	100
18208 (TTStrategyEngine)	116, 147
18210 (PriceFeedStatus)	83, 89
18211 (ContractTerm)	40, 42
18212 (UnderlyingContractTerm)	45, 47
18214 (IncludeNumberOfOrders)	80
18216 (ExchCred)	148
19 (ExecRefID)	107, 146
198 (SecondaryOrderID)	108, 126
20 (ExecTransType)	107, 146
200 (MaturityMonthYear)	39, 41
201 (PutOrCall)	40, 42
202 (StrikePrice)	40, 42
204 (CustomerOrFirm)	50, 52
205 (MaturityDay)	39, 40, 41, 42
206 (OptAttribute)	40, 42
207 (SecurityExchange)	38, 39, 41, 97, 159, 206
21 (HandInst)	108, 121, 130, 147
210 (MaxShow)	112, 122, 131, 142
262 (MDReqID)	79, 82, 83, 89
263 (SubscriptionRequestType)	79, 102, 139, 158
264 (MarketDepth)	79
265 (MDUpdateType)	79
266 (AggregatedBook)	79
267 (NoMDEntryTypes)	80
268 (NoMDEntries)	84, 89
269 (MDEntryType)	80, 84, 90
270 (MDEntryPx)	84, 90
271 (MDEntrySize)	85, 91
277 TradeConnection	87, 93, 115
279 (MDUpdateAction)	89
288 MDEntryBuyer	93
289 MDEntrySeller	93
290 (MDEntryPositionNo)	86, 92
308 (UnderlyingSecurityExchange)	44, 46
309 (UnderlyingSecurityID)	47
31 (LastPx)	115, 145
310 (UnderlyingSecurityType)	44, 46
311 (UnderlyingSymbol)	44, 46
313 (UnderlyingMaturityMonthYear)	45, 46
314 (UnderlyingMaturityDay)	45, 47
315 (UnderlyingPutOrCall)	45, 47
316 (UnderlyingStrikePrice)	45, 47
317 (UnderlyingOptAttribute)	45, 47
318 (UnderlyingCurrency)	47
319 (RatioQty)	46, 48
32 (LastShares)	110, 145
320 (SecurityReqID)	96, 98
321 (SecurityRequestType)	96
322 (SecurityResponseID)	98
323 (SecurityResponseType)	98
324 (SecurityStatusReqID)	102, 103

E TT Trading Networks

325 (UnsolicitedIndicator)	159
326 (SecurityTradingStatus)	103
34 (MsgSeqNum)	31, 33
346 (NumberOfOrders)	87, 93
35 (MsgType)	31, 33
36 (NewSeqNo)	71
37 (OrderID)	108, 126, 129, 134, 136, 146
371 (RefTagID)	72
372 (RefMsgType)	72, 157
373 (SessionRejectReason)	72
375 (ContraBroker)	114, 144
378 (ExecRestatementReason)	114
379 (BusinessRejectRefID)	157
38 (OrderQty)	112, 122, 130, 142
380 (BusinessRejectReason)	157
387 (TotalVolumeTraded)	83, 92
39 (OrderStatus)	110, 127
393 (TotalNumSecurities)	98
40 (OrdType)	113, 123, 132, 143
41 (OrigClOrdID)	109, 126, 129, 134
43 (PossDupFlag)	31, 34
432 (ExpireDate)	114, 124, 131, 144
434 (CxlRejResponseTo)	127
44 (Price)	111, 122, 130, 142
440 (ClearingAccount)	49, 51
442 (MultiLegReportingType)	110, 147
45 (RefSeqNum)	72, 157
47 (Rule80A)	50, 52
48 (SecurityID)	38, 43, 97
49 (SenderCompID)	31, 33
50 (SenderSubID)	32, 34
52 (SendingTime)	32, 34
54 (Side)	122, 131
55 (SecurityID)	38, 43
55 (Symbol)	38, 39, 41, 96
56 (TargetCompID)	31, 33
57 (TargetSubID)	32, 33, 34
58 (Text)	103, 111, 127, 147, 160
59 (TimeInForce)	114, 124, 131, 144
6 (AvgPx)	115
60 (TransactTime)	115, 121, 127, 129, 147
6038 (OrderEnteredTime)	115
63 (SettlType)	40, 42
7 (BeginSeqNo)	70
75 (TradeDate)	114, 144
76 (ExecBroker)	114, 144
77 (OpenClose)	113, 124, 131, 143
8 (BeginString)	31, 32
855 (Symbol)	38, 39, 41
864 (NoEvents)	100
865 (EventType)	100
866 (EventDate)	100
9 (BodyLength)	31, 33
96 (RawData)	65
97 (PossResendFlag)	34
98 (EncryptMethod)	65
99 (StopPrice)	112, 122, 130, 142
TargetCompID tag	31, 33
TargetSubID configuration parameter	32
TargetSubID tag	32, 33, 34
TestReqId tag	69

Text tag	103, 111, 127, 147, 160
TickSize tag	99
TimeInForce tag	114, 124, 131, 144
TotalNumOrders tag	111
TotalNumPosReports tag	145
TotalNumSecurities tag	98
TotalVolumeTraded tag	83, 92
TradeConnection tag	87, 93, 115
TradeDate tag	114, 144
Trader component block	49
Trader IDs, CME and CBOT regulations	35
TradesInFlow tag	100
TransactTime tag	115, 121, 127, 129, 147
TrdType tag	117, 148
TT Gateways	
supported	14
TTAccountType tag	50, 52
TTOrderTagData tag	116, 125, 133, 147
TTSecurityAlias tag	100
TTSessionID tag	115, 147
TTStrategyEngine tag	116, 147
TTUserTagData tag	116, 125, 133, 147

U

Underlying instruments	44
UnderlyingContractTerm tag	45, 47
UnderlyingCurrency tag	47
UnderlyingMaturityDay tag	45, 47
UnderlyingMaturityMonthYear tag	45, 46
UnderlyingOptAttribute tag	45, 47
UnderlyingPutOrCall tag	45, 47
UnderlyingSecurityAltID tag	47
UnderlyingSecurityExchange tag	44, 46
UnderlyingSecurityID tag	47
UnderlyingSecurityType tag	44, 46
UnderlyingStrikePrice tag	45, 47
UnderlyingSymbol tag	44, 46
UnsolicitedIndicator tag	159
Username tag	53, 126

V

Version, FIX protocol	31, 32
-----------------------	--------