CME Group

Market Data Platform FIX/FAST

Core Functionality

Version 2.1

9/9/10

Futures trading is not suitable for all investors, and involves the risk of loss. Futures are a leveraged investment, and because only a percentage of a contract's value is required to trade, it is possible to lose more than the amount of money deposited for a futures position. Therefore, traders should only use funds that they can afford to lose without affecting their lifestyles. And only a portion of those funds should be devoted to any one trade because they cannot expect to profit on every trade.

All references to options refer to options on futures.

CME Group is a trademark of CME Group Inc. The Globe Logo, CME, Chicago Mercantile Exchange, Globex, iLink, E-mini, CME EOS Trader, Galax-C, FirmSoft, CME DataSuite, and CME DataMine are trademarks of Chicago Mercantile Exchange Inc. New York Mercantile Exchange, NYMEX, miNY, and ClearPort are registered trademarks of the New York Mercantile Exchange, Inc. COMEX is a trademark of Commodity Exchange, Inc.

FIX™ and FAST™ are trademarks of FIX Protocol Limited. FIX/FAST℠ is a service mark of FIX Protocol Limited.

Dow Jones℠, Dow Jones AIG Commodity Index℠, The Dow℠, Dow Jones Industrial Average℠, and DJIA℠ are service marks of Dow Jones & Company, Inc. and American International Group, Inc. (AIG) and have been licensed for use for certain purposes by the Board of Trade of the City of Chicago, Inc (CBOT®). CBOT futures and options on futures contracts based on the Dow Jones Industrial Average℠ are not sponsored, endorsed, sold or promoted by Dow Jones℠, and Dow Jones℠ makes no representation regarding the advisability of trading such product(s).

BM&FBOVESPA™ is a trademark of BM&FBOVESPA, KRX™ is a trademark of Korea Exchange, DME™ is a trademark of Dubai Mercantile Exchange, BMD™ is a trademark of Bursa Malaysia, BMV™ is a trademark of Bolsa Mexicana De Valores.

All other trademarks are the property of their respective owners.

The information within this document has been compiled by CME Group for general purposes only. CME Group assumes no responsibility for any errors or omissions. Additionally, all examples in this brochure are hypothetical situations, used for explanation purposes only, and should not be considered investment advice or the results of actual market experience.

All matters pertaining to rules and specifications herein are made subject to and are superseded by official CME, CBOT, and NYMEX rules. Current rules should be consulted in all cases concerning contract specifications.

Copyright © 2010 CME Group Inc. All rights reserved.

1. Introduction

This document contains information on core features and functionality for Market Data Platform FIX/FAST.

Refer to the following sections for detailed information:

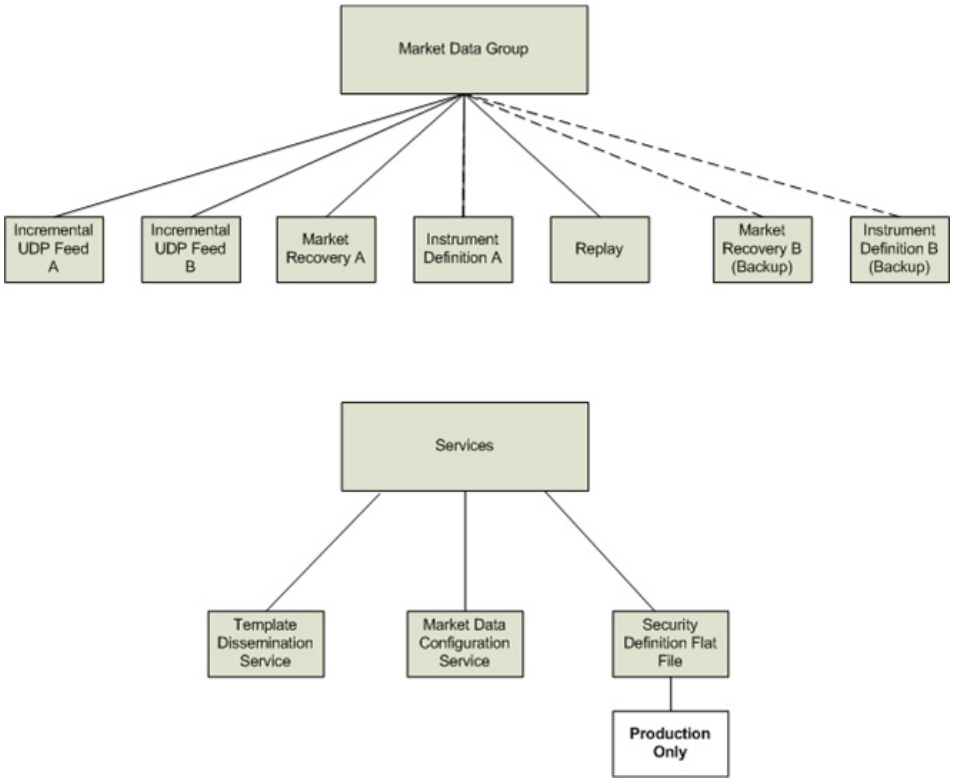
2. Architecture

This section contains an architecture overview for FIX/FAST on the Market Data Platform.

2.1 System Architecture Overview

This section contains an overview of the MDP environment.

Figure 1. System Architecture



2.1.1 Market Data Group

Market Data Platform FIX/FAST architecture is explained in this section in terms of a market data group. A market data group is a set of UDP channels used to produce market data messages for a set of instruments.

2.1.1.1 Incremental UDP Feed A and B

UDP Feed A and UDP Feed B are used to disseminate CME Group incremental market data using bandwidth efficient FAST encrypted FIX messages. All FIX message types (e.g., book update, statistics, quotes, instrument definitions, trades, instrument / instrument group, security definitions, CME Globex status) are sent through both UDP Feed A and UDP Feed B applicable Market Data Groups. This duality minimizes the chance of message loss due to UDP. Each FIX message is sent down both feeds. Please note that a single FIX/FAST Market Data Incremental Refresh (tag 35-MsgType=X) message can contain multiple updates for multiple instruments.

Note: **UDP Feed A and UDP Feed B should be used for arbitration.**

2.1.1.2 Market Recovery (UDP)

Market Recovery (UDP) Feed A is used to disseminate CME Group market data snapshots for all books with any activity since the beginning of the week. Market Recovery (UDP) Feed B functions as a backup in the event that Feed A becomes inoperable. Market Recovery (UDP) Feed A and Market Recovery (UDP) Feed B should not be used for arbitration. Book updates (outright and implied) and statistics are sent through the applicable incremental market data channel. A single FIX/FAST message contains the market state for a given instrument. Snapshots are replayed at a constant flow of configurable TPS. Expired instruments are included on the Market Recovery feed.

Note: **CME strongly recommends that the Market Recovery feeds be used for recovery purposes only. Once client systems have retrieved recovery data, client systems should stop listening to the Market Recovery feeds.**

2.1.1.3 Instrument Definition (UDP)

Instrument Definition (UDP) Feed A is used to disseminate CME Group instrument definitions. A single FIX/FAST message contains the definition of a given instrument. Instrument definitions are replayed at a constant flow of configurable TPS. The link can be found at http://www.cmegroup.com/globex/files/InstrumentDefinitionMarketRecoveryRates.pdf. Expired instruments are not included on the Instrument Definition channel. Instrument Definition (UDP) Feed B functions as a backup in the event that Feed A becomes inoperable.

Note: **Instrument Definition (UDP) Feed A and Instrument Definition (UDP) Feed B should not be used for arbitration.**

2.1.1.4 Replay (TCP - Historical)

The TCP historical replay component allows you to request a replay of a set of messages already published on the UDP Incremental Market Data Channel or a snapshot message. The request specifies messages to be replayed. The request uses the FIX Market Data Request message (35=V).

This type of request is sent through a new TCP connection established by the customer. The responses are sent by CME Group through this same connection and the connection is then closed by CME Group once the resend is complete. All responses are FIX/FAST encoded (including the reject response). Replay is limited to a maximum of 2500 messages, the recommended request level.

Note: **This is not the preferred method for recovery (use market recovery).**

2.1.2 Services - Template Dissemination, Market Data Configuration, and Security Definition Flat File

There are three services available: Template Dissemination Service, Market Data Configuration Service, and Security Definition Flat File.

Service	Description
Template Dissemination	FIX/FAST is a template based protocol. As a result, messages can only be interpreted using a template. Each message contains a unique Template ID that references the template to use to interpret the message. The template dissemination service provides a method for client systems to receive all of the CME active templates, or the templates associated with either a Template ID or a Market Data Group. For additional information on this, refer to “Template Distribution” on Page 36.
Market Data Configuration	The Market Data Configuration Service allows clients systems to receive the list of all Market Data Channel configurations (multicast IP, product group, Security ID, and sub-channel identifier). An FTP site is used for this service.
Security Definition Flat File	The Security Definition Flat File (secdef.dat) is available only for the Production environment and is generated each evening from Sunday through Thursday at 2:30pm and 8:30pm Central time. It contains the Security Definition Tag 35-MsgType=d message information for the start of the next trading day, including instrument data and daily settlement prices. This flat file is not refreshed throughout the trading day. Once a trading day has started, the following intra-day instrument changes are not reflected in the file until a new one is generated that evening: • New instruments added during the day • Covered options strategies that expire at the end of the trading day • User-Defined Spreads (UDS) created during the trading day ***Note:*** **CME Group strongly recommends using this file either in addition to or as a back up for the market data channels due to risk of unaccounted for intraday instrument additions and deletions.**

2.1.2.1 FTP Site Information - Template Dissemination and Market Data Configuration

An FTP site (ftp://ftp.cmegroup.com) is used to disseminate templates and market data configuration information. This FTP site contains the template files and configuration files for all environments and the Security Definition Flat File for the Production environment. Template and market data configuration details for the production environment are only available to customers after the certification process is complete.

Information applies as follows in the table:

- Environment - specific environment (i.e., Certification, New Release, Production).

Note: **The AutoCert+ tool will indicate which environment you need to connect to for certification. For additional information on AutoCert+ access, refer to:** AutoCert+ Access Guide

- Service - the Template or Configuration service.
- FTP Site - address of FTP site.
- Directory Location - identifies the directory.
- Client System Update Schedule - Client systems should download updates according to the schedule specified.

Environment	Service	FTP Site	Directory Location	Client System Update Schedule
Certification	Template	ftp.cmegroup.com	/Cert/Templates	Sunday prior to market open
Certification	Configuration	ftp.cmegroup.com	/Cert/Configuration	daily
Certification AutoCert+	Template	ftp.cmegroup.com	/CertAutoCertPlus/Templates	Sunday prior to market open
Certification AutoCert+	Configuration	ftp.cmegroup.com	/CertAutoCertPlus/Configuration	daily
New Release Certification	Template	ftp.cmegroup.com	/NRCert/Templates	Sunday prior to market open
New Release Certification	Configuration	ftp.cmegroup.com	/NRCert/Configuration	daily
New Release Certification AutoCert+	Template	ftp.cmegroup.com	/NRAutoCertPlus/Templates	Sunday prior to market open
New Release Certification AutoCert+	Configuration	ftp.cmegroup.com	/NRAutoCertPlus/Configuration	daily
Production	Template	ftp.cmegroup.com	/Production/Templates	Sunday prior to market open
Production	Configuration	ftp.cmegroup.com	/Production/Configuration	daily
Production	Flat File	ftp.cmegroup.com	/Production	8:30pm Sunday - Thursday

Note: **There is also an ftp site on the CME Group network that can be accessed as an alternative to the public ftp.cmegroup.com site. Please contact your account manager for additional information.**

2.2 System Startup

This section contains an overview of the Sunday startup process for the production environment.

2.2.1 Pre-Opening Startup

For a startup prior to the weekly market open, all market data including book updates, statistics, quotes, instrument definitions, trades, instrument/instrument group, status) will be disseminated through the Incremental UDP Feed A and Feed B. Book updates include bid/ask/trade. Statistics include high trade, low trade, volume, and settlement. Instrument definitions include the list of all tradable instruments. Follow the process below to ensure that all necessary market data is received:

1. Download the configuration files and template files from the ftp site. Refer to “Services - Template Dissemination, Market Data Configuration, and Security Definition Flat File” on Page 9 for more information.
2. Listen to the Incremental feed for incremental market data and start normal processing.

2.2.2 Late Joiner Startup

For a late joiner startup, follow the process below to ensure that all necessary market data is received:

1. Download the configuration files and template files from the ftp site. Refer to “Services - Template Dissemination, Market Data Configuration, and Security Definition Flat File” on Page 9 for more information.
2. Listen to the Instrument Definition feed.

   Refer to “Instrument Replay Overview” on Page 104 for more information.
3. Listen to the Incremental feed for incremental market data. Begin the natural refresh process and begin queuing messages.

Note: **The incremental market data may complete a natural refresh (liquid instruments only) that would construct the current, correct state of a book. Refer to “Natural Refresh” on Page 109 for more information.**

4. Listen to the Market Recovery feed for the latest snapshots.

Use the latest snapshot to verify that the book was correctly created via natural refresh and to retrieve the latest statistics. When the latest snapshots are received, the value for tag 369-LastMsgSeqNumProcessed in the snapshot matches tag 34-MsgSeqNum in the incremental feed message. Also, the value for tag 83-RptSeq in the snapshot matches tag 83-RptSeq in the incremental feed.

If the book for an instrument was not completely constructed using natural refresh, then apply the snapshot. Start discarding queued messages from the incremental feed until tag 34-MsgSeqNum in the message has the same value as tag 369-LastMsgSeqNumProcessed in the snapshot. The discarded messages contain information that was already included in the snapshot message.

Note: **Information for instruments included for the first time in the latest incremental feed message (with 34-MsgSeqNum equal to tag 369-LastMsgSeqNumProcessed in the snapshot) may not be in the latest snapshot. Apply the latest incremental feed message to obtain this information.**

5. Stop listening to the Market Recovery and Instrument Definition feeds.
6. Start normal processing.

2.2.3 Heartbeats

When the Market Recovery and Instrument Definition components begin, CME Globex sends Heartbeat (tag 35-MsgType=0) messages at a predefined interval on the Market Recovery and Instrument Definition feed. The components begin heartbeating on Sunday from the time they start until the time the first data is sent. When the Market Recovery and Instrument Definition channels start looping, CME Globex no longer sends Heartbeat tag 35-MsgTpe=0 messages.

2.3 Incremental Feed Arbitration

CME Group strongly recommends that client systems process both the Incremental Feed A and Incremental Feed B due to the unreliable nature of UDP transport.

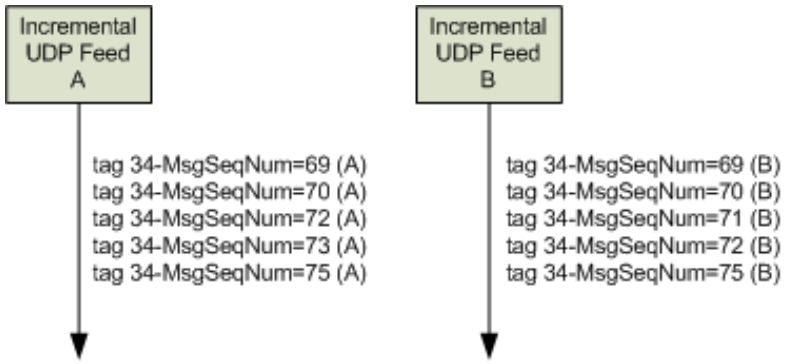
Messages can come first on the Incremental Feed A or Incremental Feed B at any time during the day. This is dependent on the network carriers and thus is not controlled by CME Group.

The Preamble sequence number, or tag 34-MsgSeqNum, can be used during the arbitration process to determine whether any messages have been missed without decoding the FAST message. Using the Preamble sequence number requires less processing than using tag 34-MsgSeqNum.

Client systems should use the A and B Incremental Feeds to arbitrate as follows:

1. Listen to messages from both Incremental Feed A and Incremental Feed B.
2. Process messages by incremental sequence number (tag 34-MsgSeqNum).
3. Discard messages if a given sequence number (tag 34-MsgSeqNum) has already been process.
4. If a sequence number (tag 34-MsgSeqNum) gap is detected - this would indicate a packet was lost on both the Incremental Feed A and Incremental Feed B. Client systems would need to initiate the Recovery process (refer to “Recovery” on Page 99 for additional information).

EXAMPLE:



Processing of messages is as follows:

Messages are received at client system from Incremental Feed A and Incremental Feed B individually:

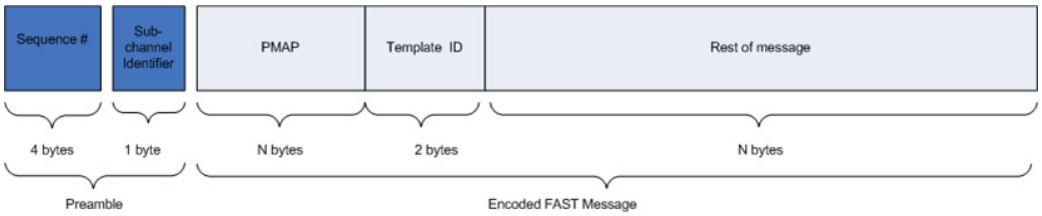
1. Receive message on feed A - tag 34-MsgSeqNum=69.
   - Process message on feed A - tag 34-MsgSeqNum=69.
2. Receive message on feed B - tag 34-MsgSeqNum=69.
   - Discard message on feed B - tag 34-MsgSeqNum=69.
3. Receive message on feed A - tag 34-MsgSeqNum=70.

- Process message on feed A - tag 34-MsgSeqNum=70.

4. Receive message on feed B - tag 34-MsgSeqNum=70.
   - Discard message on feed B - tag 34-MsgSeqNum=70.
5. Receive message on feed B - tag 34-MsgSeqNum=71.
   - Process message on feed B - tag 34-MsgSeqNum=71.
6. Receive message on feed B - tag 34-MsgSeqNum=72.
   - Process message on feed B - tag 34-MsgSeqNum=72.
7. Receive message on feed A - tag 34-MsgSeqNum=72.
   - Discard message on feed A - tag 34-MsgSeqNum=72.
8. Receive message on feed A - tag 34-MsgSeqNum=73.
   - Process message on feed A - tag 34-MsgSeqNum=73.
9. Receive message on feed A - tag 34-MsgSeqNum=75.
   - Discard message on feed A - tag 34-MsgSeqNum=75.
10. Receive message on feed B - tag 34-MsgSeqNum=75.
    - Discard message on feed B - tag 34-MsgSeqNum=75.
    - Gap detected because last message processed was tag 34-MsgSeqNum=73 on both feed A and feed B.
    - Begin recovery process - tag 34-MsgSeqNum=75.

2.4 FIX/FAST Message Preamble

The Preamble provides the sequence number and sub-channel identifier for the related FAST message. The Preamble consists of 5 non-FAST encoded bytes in Big Endian format that are found before all FIX/FAST messages on all feeds (Incremental, Market Recovery, Instrument Definition, and TCP Replay). The Preamble is found before the FAST encoded message, and contains the sequence number and sub-channel identifier. Processing of the Preamble is optional and FAST messages are not impacted by it.



2.4.1 Preamble Sequence Number

The Preamble sequence number is a 4 byte, unsigned integer representing the value distributed in tag 34-SeqNum of the related FAST message. The Preamble sequence number can be used during the arbitration process to determine whether any messages have been missed without decoding the FAST message, reducing processing time.

2.4.2 Preamble Sub-Channel Identifier

The sub-channel identifier is 1 unsigned byte found between the Preamble sequence number and the FAST message. It will be used to identify the sub-channel for all FAST messages. This makes it possible to process only the messages which carry market data for certain sub-channels and ignore all other sub-channels.

The sub-channel identifiers are mapped to group codes in the **config.xml** file.

3. FAST Implementation

This section describes how to implement FIX Adapted for STreaming (FAST) protocol.

3.1 Introduction

The FIX Adapted for STreaming (FAST) Protocol has been developed as part of the FIX Market Data Optimization Working Group. FAST is designed to optimize electronic exchange of financial data, particularly for high volume, low latency data dissemination. This document describes implementation of FAST in receiving and processing CME Group FIX/FAST-encoded electronic market data feed.

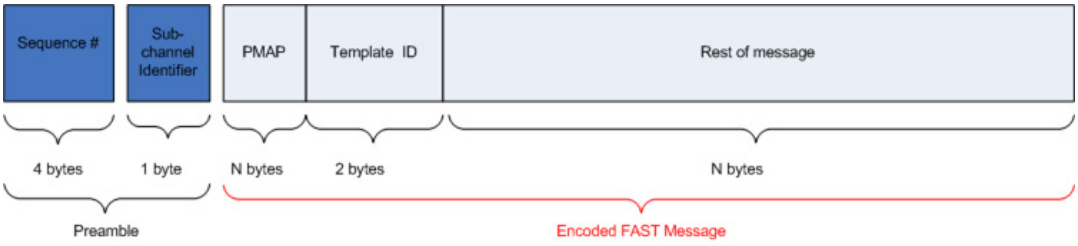
For more information see the FAST specification (version 1.x.1) at:

http://www.fixprotocol.org/documents/3066/FAST%20Specification%201%20x%201.pdf

FAST is a data compression algorithm that significantly reduces bandwidth requirements and latency between sender and receiver. FAST works especially well at improving performance during periods of peak message rates. FAST extends the base FIX specification and assumes the use of FIX message formats and data structures. FAST is a standalone specification that uses templates to inform the receiver which operations to use in decoding. Templates allow FAST to achieve high levels of data compression with low processing overhead and latency compared to other compression utilities such as Zlib.

The Preamble is found before the FAST encoded message, and contains the sequence number and sub-channel identifier.

Note: **For more information on the Preamble, refer to "FIX/FAST Message Preamble" on Page 13.**



Note: **This document describes concepts applicable to CME Group-specific FAST implementation; this document is supplementary to the FAST specification referenced above.**

3.1.1 Stop Bit Encoding

Stop Bit encoding is a process incorporated in FAST that eliminates redundancy at the data field level by using a stop bit instead of the traditional separator byte. In FAST, a stop bit is used instead of FIX's traditional <SOH> separator byte. Thus 7 bits of each byte are used to transmit data and the eighth bit is used to indicate the end of a field.

3.1.1.1 FAST 7-Bit Binary Representation

FAST renders numbers into binary across the 7 data bits in each byte. Thus a number equal to or less than 2^7-1, (127) occupies one byte, a number between 2^7 and 2^7*2 – 1 (16,383), occupies two bytes, etc.

3.1.2 Implicit Tagging

In traditional FIX messages each field takes the form “Tag=Value<SOH>” where the tag is a number representing which field is being transmitted and the value is the actual data content. The ascii <SOH> character is used as a byte delimiter to terminate the field. For example:

35=x|268=3 (message header)

279=0|269=2|270=9462.50|271=5|48=800123|22=8 (trade)

279=0|269=0|270=9462.00|271=175|1023=1|48=800123|22=8|346=15 (new bid 1)

279=0|269=0|270=9461.50|271=133|1023=2|48=800123|22=8|346=12 (new bid 2)

FAST eliminates redundancy with a template that describes the message structure. This technique is known as implicit tagging as the FIX tags become implicit in the data. A FAST template replaces the tag=value syntax with “implicit tagging” as follows:

- tag numbers are not present in the message but specified in the template
- fields in a message occur in the same sequence as tags in the template
- the template specifies an ordered set of fields with operators

3.1.3 Field Encoding Operators

FAST functions as a state machine and must know which field values to keep in memory. FAST compares the current value of a field to the prior value of that field and determines if the new value should be constant, default, copy, delta (integer or string), increment, or tail.

3.1.3.1 Dictionary Context

CME Group uses a dictionary context on a per-packet basis. A dictionary is a cache in which previous values are maintained. **All dictionary entries are reset to the initial values specified after each UDP packet. Currently, CME Group sends one message per UDP packet.**

3.1.3.2 Field Operators

Note: **The following are general descriptions and may have exceptions depending upon the scenario.**

A field within a FAST template will generally have one of the Field Operators described below indicating the required decoding action. Please note that in some cases it is possible for a field to have no Field Operator.

- **Constant** – indicates that the field will always contain a predetermined value as specified by the *value* attribute. If the value is optional this field will contain a Pmap bit; if mandatory this field will not contain a Pmap bit.
- **Default** – indicates that the default value defined in the template should be used as the decoded value when a data value is not present. If a data value is present, use that value.
- **Copy** – indicates that the data value in the prior occurrence of this field should be used if a data value is not present for this occurrence.
- **Delta for integers** – when used with an integer, indicates that the data value represents the arithmetic difference between the current and prior values.

- **Increment** – indicates that the data value for the prior occurrence of this field should be incremented by 1 if a data value is not present for the current occurrence. This operator works with integers only.

3.1.4 Data Types

A field within a FAST template will have one of the following Data Types indicating the required decoding action:

- **String** – used to represent ASCII or Unicode values using the FAST 7-bit binary encoding.
- **Signed Integer** – used to represent a signed (+/-) integer using the FAST 7-bit binary encoding. A twos complement integer representation is used, with the most significant data bit being the sign bit. Note that contiguous leading bits of the same value must be dropped (so, for instance, only 1 byte is required to encode -1, 0xFF). In some cases, a 0x00 byte will be the most significant byte sent to preserve sign, so 64 is represented 0x00 0xC0.
- **Unsigned Integer** – used to represent unsigned integers using the FAST 7-bit binary encoding.
- **Decimal** – used to represent a floating point number as exponent and mantissa. The exponent is a signed integer used to express precision and the mantissa is a signed integer used to express the value. The numerical value is obtained by multiplying the mantissa with the base-10 power of the exponent expressed as:

  number = mantissa * 10^{exp}

  The exponent and mantissa can be encoded as a single, composite field or as individual conjoined fields.

3.1.5 FAST Template

A FAST template corresponds to a FIX message type and uniquely identifies an ordered collection of fields. The template also includes syntax indicating the type of field and transfer decoding to apply. A template is communicated between CME Group and client systems in XML syntax using the FAST v1.1 Template Definition Schema maintained by FIX. The XML format is human- and machine-readable and can be used for authoring and storing FAST templates. Session Control Protocol (SCP) will not be used.

3.1.6 Presence Map and Stop Bit

The Presence Map (Pmap) indicates which fields in the template have data present and which fields have data implied. A Pmap is a sequence of encoded bits with each bit representing a template field according to sequence. Fields with data present have the Pmap bit set to ‘1’. Fields with data implied have the Pmap bit set to ‘0’ (Exception: fields with a Constant operator in which the bit is set to ‘1’ for an implied state).

A Pmap occurs at the beginning of each FAST message and at the beginning of any sequence/group of fields as long as those fields are defined such that a Pmap slot is required.

3.1.6.1 Presence Map Rules

Pmap rules determine when a FAST template field requires a corresponding Pmap bit. A field does not require a bit in the pmap when it meets any of the following criteria:

1. The field is defined as mandatory without a field operator – a value will always be present.
2. The field is defined as mandatory with a constant operator – a value should always be instantiated in the decoded message based on the value in the template.
3. The field is defined as mandatory with a delta operator – a delta value is always present.

4. The field is defined as optional without a field operator – either a value or NULL will always be present.
5. The field is defined as optional with a delta field operator – a delta value or NULL will always be present.

See the FAST specification (version 1.x.1) for complete information on presence map rules.

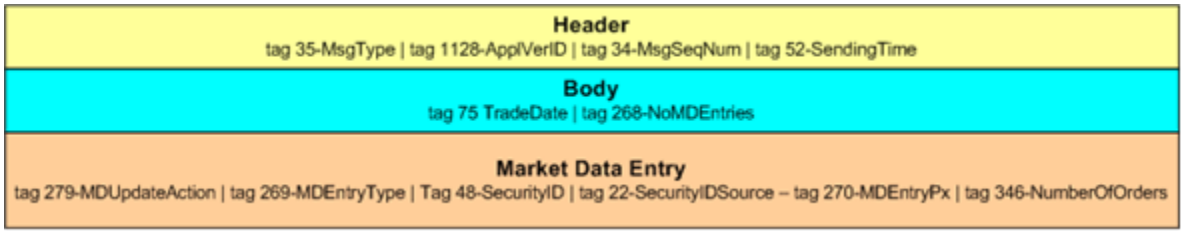
Rules for Determining if a Presence Map Bit is Required

Operation	Mandatory	Optional
None	No	No
Constant	No	Yes
Copy	Yes	Yes
Default	Yes	Yes
Delta	No	No
Increment	Yes	Yes

3.1.6.2 Message Structure

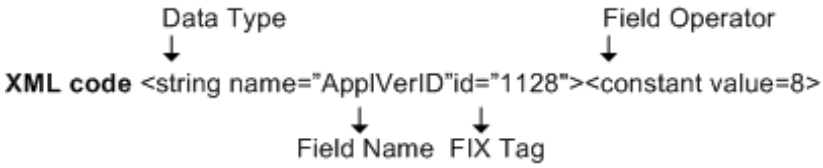
Note: **The Market Data Incremental Refresh (tag 35-MsgType = X) message is used for example purposes throughout this document; not all FAST messages follow this format.**

The FIX Market Data Incremental Refresh (tag 35-MsgType = X) message is made up of three components: a Header, a Body and a set of one or more Market Data Entries as shown in the diagram below. The **Header** carries transmission details. The **Body** carries information pertinent to all entries in the message such as TradeDate. The **Market Data Entry** carries specific instructions for updating the book or recording trades. The templates provided by CME Group conform to this general structure and use only fields that are part of the Market Data Incremental Refresh (tag 35-MsgType = X) message.



Example: How to Read a Template Field

Field Operators and Data Types appear within the Template as follows:



3.1.6.3 Template ID Usage

Each template is assigned a Template ID to uniquely describe the format of an encoded message. A Template ID is carried in every encoded message to provide a reference to the correct template for decoding purposes.

The Template ID is an unsigned integer carried as the first data field following the first Pmap of every message allowing the decoding system to apply the correct template to the message upon receiving it.

Example: Template ID

```
2   <template name="MDIncRefresh" id="35" xmlns="http://www.fixprotocol.org/ns/fast/td/1.1">
3     <typeRef name="MDIncRefresh"/>
```

CME Group will not send the same Template ID more than once in a UDP packet. When the packet is received, the sequence to begin decoding a message is:

1. Locate and read the first Pmap field, which begins in the first first byte of the application data preceding the MDP/FIX Header.
2. Treat the next field as the Template ID (the first logical field of the message).
3. Retrieve the template specified by the Template ID.
4. Decode the remainder of the message.

3.1.6.4 Optional vs. Mandatory Fields

A field is defined as optional when it is possible for a value to not be present in the decoded message. When a field is optional, the wire representation “null” indicates the field is not present. A field is defined as mandatory when that field must be present in the decoded message. Note that a field may not be present (“coded away”) in the wire-formatted message but can be present in the decoded message as the result of an encoding operation (e.g., default) that removes data for efficiency but continues to imply data is present through the Pmap.

Optional fields are useful when a generic template is used to provide multiple market data types such as book updates and trades. In this situation, there may be fields which are present in one occurrence of the repeating group but not in another within a given message.

For example, the MDEntries repeating group that is present in the MDIncRefresh template shown above can be used to express a trade, bid, ask, high, low, etc. within a single message. A trade entry will not use tag 346, NumberOfOrders, which will then be defined as optional. However, FAST requires that this field be accounted for in the encoded message if specified in the template. This is done through the use of a reserved value of NULL to indicate that the field is not present in the decoded data. In the serialization layer, FAST reserves binary zeros to indicate a NULL value which tells the decoder that no data is present for this template distribution.

Note: **CME Group requires that client systems use an API layer to load templates rather than implement hard coded templates. Since templates change, this will facilitate template modification in production environments.**

Table 3.1. Rules for Determining if a Field is Nullable

Operation	Mandatory	Optional
None	No	Yes
<Constant/>	No	No
<Copy/>	No	Yes
<Default/>	No	Yes
<Delta/>	No	Yes
<Increment/>	No	Yes

3.1.6.5 XML Template Example

A template consists of Field Instructions that define the fields contained in the message. Field Instructions specify the field name, tag number, data type, field operator, and presence attribute that indicates if a field is optional or mandatory.

A sample market data template is shown below. The syntax is standard XML and can be parsed using a variety of open source tools. Valid template syntax is determined by the FAST Template Schema which is available in the FAST v1.1 specification.

- The real-time feed templates are based on the Market Data Incremental Refresh message type.

Client FAST Decoders read the template file to recognize how each field is encoded in terms of data type representation (Transfer) and data redundancy removal (Field).

The template is constructed of several sections including Template Identification, Header, Body and Sequence. Template Identification provides the template name and identifier. The Header includes FIX header fields such as ApplVerID (tag 1128), MsgType (tag 35), and SendingTime (tag 52). The Body provides information common across all repeating groups. Sequence represents a repeating group with a corresponding length field and a set of repeating group fields which carry the detailed entry information.

```
<?xml version="1.0" encoding="UTF-8"?>
<template name="MDIncRefresh" id="30" xmlns="http://www.cme.com/mdp/fast-templates" >
  <typeRef name="MDIncRefresh"/>
  <string name="ApplVerID" id="1128"> <constant value="7"/> </string>
  <string name="MessageType" id="35"> <constant value="X"/> </string>
  <string name="SenderCompID" id="49"> <constant value="CME"/> </string>
  <uInt32 name="MsgSeqNum" id="34"> </uInt32>
  <uInt64 name="SendingTime" id="52"> <delta/> </uInt64>
  <uInt32 name="TradeDate" id="75"><copy/> </uInt32>
  <sequence name="MDEntries"><length name="NoMDEntries" id="268"></length>
    <decimal name="MDEntryPx" id="270">
      <exponent><copy value="-2"/></exponent>
      <mantissa><delta/></mantissa> </decimal>
    <int32 name="MDEntrySize" id="271"> <delta/> </int32>
    <uInt32 name="MDEntryTime" id="273"> <delta/> </uInt32>
    <uInt32 name="MDPriceLevel" id="1023" presence="optional"> <increment/> </uInt32>
    <uInt32 name="MDUpdateAction" id="279"> <copy value="1"/> </uInt32>
    <string name="MDEntryType" id="269"> <copy value="0"/> </string>
    <uInt32 name="SecurityID" id="48"> <copy/> </uInt32>
    <uInt32 name="SecurityIDSource" id="22"> <constant value="8"/> </uInt32>
    <int32 name="NumberOfOrders" id="346" presence="optional"> <delta/> </int32>
    <string name="QuoteCondition" id="276" presence="optional"> <copy value="K"/> </string>
    <string name="TickDirection" id="274" presence="optional"> <default/> </string>
    <int32 name="NetChgPrevDay" id="451" presence="optional"> <default/> </int32>
    <string name="TradeCondition" id="277" presence="optional"> <default/> </string>
    <int32 name="TradeVolume" id="1020" presence="optional"> <default/> </int32>
    <string name="TradingSessID" id="336"><default value="1"/> </string>
  </sequence>
</template>
```

Please refer to the Appendix for a decomposition of the template above with an explanation for each instruction.

Templates are available for each CME Globex environment from the ftp site at ftp.cmegroup.com.

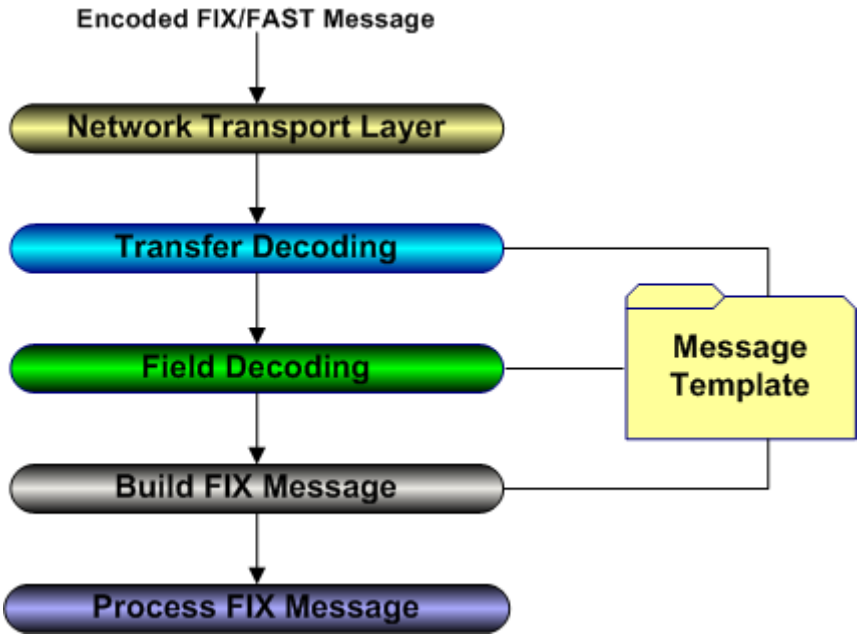
3.2 FAST Decoding

This section presents one possible approach to decoding CME Group FAST messages to demonstrate the fundamental concepts involved in the process. This approach is provided for example purposes and is not definitive.

FAST processing decodes a FIX/FAST message by means of FAST templates provided by CME Group. The FAST template contains the instructions to decode and reconstruct compressed message data into the FIX format and also supports repeating groups (sequences) that allow a single message to convey multiple instructions (i.e. book update, trade, high/low, etc.).

Note: **CME Group FAST encoding is bit-specific to the FAST message specification. CME Group FAST encoding always uses the global dictionary.**

A decoding sequence can follow the steps diagramed below; this document describes Transfer Decoding, Field Decoding, and Building the FIX message only.



Step 1. Transport - Client system receives encoded FAST message.

Step 2. Transfer Decoding

- Identify template
- Extract binary encoded bits
- Map bits to fields per template

Step 3. Field Decoding - Apply operators to determine values per template field.

Step 4. Build FIX message.

Step 5. Process FIX message.

3.3 Transfer Decoding Overview

Transfer Decoding is the initial step that converts data from the FAST 7-bit binary format.

Note: **The decoding process can take place on a field-by-field basis, de-serializing and rebuilding the decoded value for each field. Alternately, the decoding process may de-serialize the entire message and then make a second iteration over the message to rebuild the decoded values.**

The FAST transfer format is a binary representation enhanced with the following attributes:

- **Stop Bit** - Each data byte contains a stop bit to indicate whether this is the last byte of a field. The stop bit is the seventh bit of the most significant byte. A stop bit set to 1 indicates the last byte of a field. A stop bit set to zero indicates this is not last byte of the field.

- **Presence Map** - The Presence Map occurs at the start of every message or repeating group and indicates the presence or absence of individual fields. The template specifies the type of transfer decoding to employ for each field in the message (e.g. string, integer, decimal, etc.).

WARNING

FAST does not support timestamps. CME Group will convert the timestamp to an integer by removing punctuation from the UTC timestamp prior to encoding. The decoding application should convert the integer to the FIX UTC format after decoding. In addition, the timestamp header on the message packet should not be used for latency comparisons. The message must be decoded to determine the timestamp for updates.

3.3.1 Field Decoding Overview

Field decoding is the second part of the decompression process that reconstructs data values according to template-specified operations. Field decoding operations are assigned per field within the template; decoding reinstates data as indicated by the template.

3.3.1.1 When to Reset Decoder State

The state of the decoder must be reset for each received UDP packet and the dictionary applied. Resetting state means that all fields are set to their initial pre-processing state. This is because UDP transport is not reliable and data in a packet cannot be dependent on data in a previous packet since it is possible for packets to be lost or arrive out of sequence.

3.3.2 Receiving Data over MDP

Currently, CME Group will send only a single FIX message per packet; however, in future implementations CME Group may send multiple messages per packet. The basic operation requires the decoding of data within a discreet packet as well as the ability to determine the end of one message and beginning of another within that packet.

3.3.2.1 Decoding a Packet

Decoding of the datagram should be conducted as a unit of work in which only the data in a given packet participates in the scope of the decoding process. To determine when the process has fully decoded a discreet message, it is important to use the template to step through a message field-by-field.

3.3.2.2 Decoding Messages in a Packet

When the decoder determines that it has processed all repeating groups in the message, and that there are no further fields specified by the template, then it can conclude that the end of the message has been reached. In this way, it is necessary to use the presence maps, NoMDEntries, and the template to determine when end of message has been reached. To recap, the steps involved are:

- Decode fields in the non-repeating body of the message using the high-level presence map.
- If the last field in the template has been reached then end of message has been encountered.

- If the NoMDEntries field is present, then process each repeating group using the repeating group presence map and template.
- When number of repeating groups processed equals the count specified in NoMDEntries, then the end of the repeating groups has been encountered.

3.3.2.3 Error Handling in a Broadcast Environment

If FAST encounters an error during the process of decoding the contents of a UDP packet, the process should stop, throw an exception, and decode the next packet. In all likelihood, the error is due to an inconsistency between the encoded data and the template being used to decode the data. At this point, it is advised that parties synchronize templates in order to ensure that the same data definitions are being used.

3.4 Decoding Sequence

This section provides a detailed example of the process for decoding a FAST message. You should thoroughly test your application to determine that you have properly decoded a message.

Note: **This decoding example is a generic description of the basic decoding process; it does not reflect the structure of the reference code.**

FAST message decoding consists of three major steps:

Step 1. Transfer Decoding

- Identify Template ID and Pmap bit value for each given field. Pmap values are 0, 1.
- Decode FAST 7-bit binary values to identify data present in the message.

Step 2. Field Decoding

- Apply field operators to extracted binary values.
- Determine state of field to be decoded.

Step 3. Build FIX Message

- Apply FIX or internal structure to decoded message

The decoding sequence in the example used throughout this section uses a FAST message containing a header, trade, and two new bids.

3.4.1 Transfer Decoding

Upon receipt of the FAST message, the client decoder loads the template according to the Template ID. From this template the client decoder identifies how to decode each field, whether a field will have a Pmap bit, and if a field can carry a null value. In this example the MDIncRefresh template contains a header and repeating group structure (italics).

Note: **The following template is an EXAMPLE TEMPLATE.**

```
<template name="MDIncRefresh" id="30">
  <typeRef name="MDIncRefresh"/>
<string name="MessageType" id="35"> <constant value="X"/> </string>
<sequence name="MDEntries"><length name="NoMDEntries" id="268"></length>
  <uInt32 name="MDUpdateAction" id="279"> <copy value="0"/> </uInt32>
  <uInt32 name="MDEntryType" id="269"> <copy value="0"/> </uInt32>
  <decimal name="MDEntryPx" id="270">
    <exponent><copy value="-2"/></exponent>
    <mantissa><delta/></mantissa> </decimal>
<int32 name="MDEntrySize" id="271"> <delta/> </int32>
<uInt32 name="MDPriceLevel" id="1023" presence="optional"> <increment/></uInt32>
<uInt name="SecurityIDSource" ID="22"><constant value="8"/></uInt32>
<int32 name="NumberOfOrders" id="346" presence="optional"> <delta/> </int32>
  </sequence>
</template>
```

3.4.2 Decoding Process

The client system reads the Pmap to identify required field-level decoding.

Pmap 1	Header	Pmap 2	Trade	Pmap 3	New Bid 1	Pmap 4	New Bid 2
11000000	Fields	**1**0101100	Fields	**1**0101000	Fields	**1**0000000	Fields

In this example light grey values indicate an unused bit.

Step 1. **Extract Pmap1 from data.** Pmap1 is the top-level Pmap that contains the Pmap bits for all fields requiring a Pmap bit in the message.

- The first bit in Pmap1 after the stop bit (in bold) is assigned to the Template ID, which always has a bit. The bit is turned on indicating that the Template ID is present in the encoded message and that the corresponding template must be loaded.
- The first field in the template is the MessageType, defined as a mandatory field using a Constant operator with a value of “X” retrieved from the template and therefore does not have a bit.
- The next field in the template is NoMDEntries. The length specified in the sequence determines the number of repeating groups in the message. This field does not have a bit since the field is mandatory and does not have an operator.
- **The remaining bits in Pmap1 are defaulted to 0 but are not referenced by the decoding process; at this point, the Header fields have been decoded.**

Step 2. **Extract Pmap2 from data.** Pmap2 represents the first repeating group and contains trade-related fields.

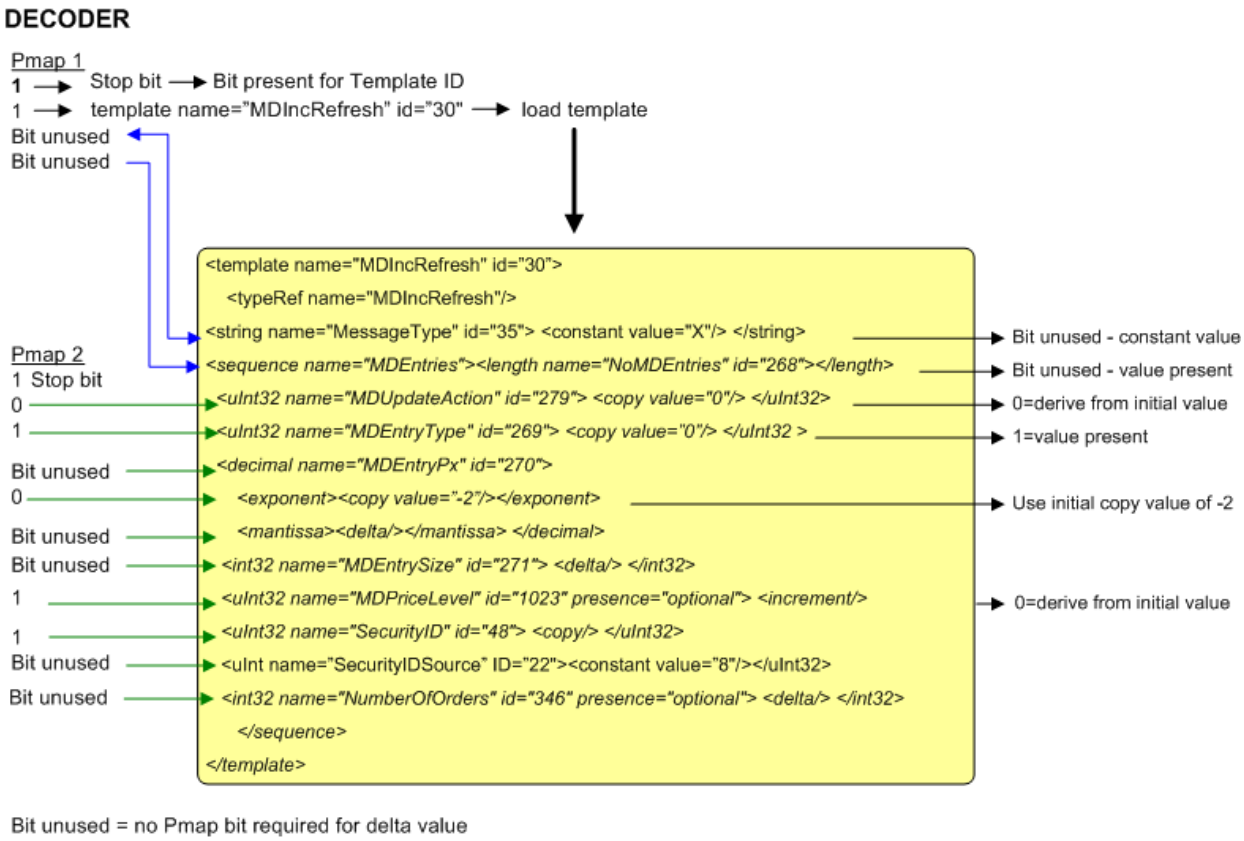
- The next field in the template is MDUpdateAction. By definition, this field requires a bit in the Pmap since it uses the *Copy* field operator. In this example, the bit is set to 0 indicating that no data is present and that the value must be derived from the initial value.
- The next field in the template is MDEntryType, which is a Copy field. The Pmap set to ‘1’ indicates an encoded value present in the field.
- The decoding process continues using the template and Pmap2 to traverse the data.

Step 3. **Extract Pmap3 from data.** Pmap3 represents the presence map for repeating group2 followed by the New Bid1 fields.

- By the time the decoding process reaches Pmap3, prior values have been established for several fields based on the content of the prior repeating group.
- The decoding process uses the prior values in combination with the encoded data to generate the proper decoded values.

Step 4. **Extract Pmap4 from data.** Pmap4 is the final presence map and the decoder recognizes that all bits are set to off indicating that either prior values should be used or that fields do not use a bit.

The following diagram illustrates the decoding process flow:



3.4.3 Build the FIX Message

The table in this example shows the step-by-step conversion for each field of the FAST message into the FIX format.

The following steps describe the actions taken to obtain the data shown in the Message Decoding Process table.

Step 1. Decoding begins with the identification of the Pmap bit for each field.

Step 2. The encoded FAST 7-bit binary values are obtained as shown in the “Encoded FAST 7-Bit Binary Value” column.

Step 3. The encoded FAST 7-bit binary values from step 2 are deserialized based on the data type specified in the template.

Step 4. The decoder maintains the state of prior values for each field throughout decoding and applies them for fields having operators of Delta, Copy, or Increment.

Step 5. Obtain fully decoded values.

Message Decoding Process Table									
				Step 1	**Step 2**	**Step 3**	**Step 4**		**Step 5**
#	**Field Name**	**Presence***	**Data Type/ Field Operator**	**Pmap Bit**	**Encoded FAST Hex/ Binary Value**	**Deserialized Encoded Value**	**Prior Value**	**Initial Value**	**Decoded Value**
BIT STREAM = 11000000 10011110 10000011									
1	Template ID	M	uint32/ copy	1	0x9e – **1**0011110	30			TID = 30
2	MsgType	M	string/ constant	No bit	None	None		X	35=X
3	NoMDEnt	M	uint32/no operator	No bit	0x83 - **1**0000011	3			268=3
Repeating Group 1 - Trade									
BIT STREAM = 10101100 10000010 00111001 01100000 11001010 10000101 10000000 00110000 01101010 11111011 10000000									
4	UpdateAc-tion	M	uint32/ copy	0	None	None		0	279=0
5	EntryType	M	uint/copy	1	0x82 - **1**0000010	2		0	269=2
6	EntryPrice	M	Exp: int32/copy Mant: int64/ delta	Exp - 0 Mant – No bit	Exp: None Mant: 0x39 0x60 0xca – 00111001 01100000 **1**1001010	Exp = None Mantissa = 946250		-2	270=9462.50
7	EntrySize	M	int32/ delta	No bit	0x85 - **1**0000101	5			271=5
8	PriceLevel	O	uint32/ increment	1	0x80 – **1**0000000	NULL		1	1023=No Value
9	SecurityID	M	uint32/ copy	1	0x30 0x6a 0xfb - **0**0110000 **0**1101010 **1**1111011	800123			48=800123
10	Security-IDSource	M	uint32/ constant	No bit	None	None		8	22=8
11	NoOrders	O	int32/ delta	No bit	0x80 – **1**0000000	NULL		0	346=No Value
Repeating Group 2 – New Bid 1									
BIT STREAM = 10101000 10000000 11001110 00000001 10101010 10000010 10010000									
12	UpdateAc-tion	M	uint32/ copy	0	None	None	0		279=0
13	EntryType	M	uint/copy	1	0x80 - **1**0000000	0	2		269=0
14	EntryPrice	M	Exp: int32/copy Mant: int64/ delta	0 No bit	Exp: None Mant: 0xce - **1**1001110	Exp = None Mantissa = -50	-2 946250		270=9462.00
15	EntrySize	M	int32/ delta	No bit	0x01 0xaa – **0**0000001 **1**0101010	170	5		271=175
16	PriceLevel	O	uint32/ increment	1	0x82 - **1**0000010	2*	NULL		1023=1

17	SecurityID	M	uint32/ copy	0	None	None	800123		48=800123
18	Security-IDSource	M	uint32/ constant	No bit	None	None	8		22=8
19	NoOrders	O	int32/ delta	No bit	0x90 - **1**0010000	16*	0**		346=15
Repeating Group 3 – New Bid 2									
BIT STREAM = 10000000 11001110 11010110 11111101									
20	UpdateAc-tion	M	uint32/ copy	0	None	None	0		279=0
21	EntryType	M	uint/copy	0	None	None	0		269=0
22	EntryPrice	M	Exp: int32/copy Mant: int64/ delta	0 No bit	Exp: None Mant: 0xce - **1**1001110	Exp = None Mantissa = -50	-2 946200		270=9461.50
23	EntrySize	M	int32/ delta	No bit	0xd6 – **1**1010110	-42	175		271=133
24	PriceLevel	O	uint32/ increment	0	None	None	1		1023=2
25	SecurityID	M	uint32/ copy	0	None	None	800123		48=800123
26	Security-IDSource	M	uint32/ constant	No bit	None	None	8		22=8
27	NoOrders	O	int32/ delta	No bit	0xfd - **1**1111101	-3	15		346=12

*M=mandatory O=optional

**Subtract 1 from non-negative optional integer fields

***For delta fields, if a null is received, the prior value in the dictionary is not changed. In this case, the intital value was not specified in the template, so the initial value is set to ‘0’. The first encoded field received has a value of null, so the prior value is ‘0’ when decoding the second instance of this field.

3.4.3.1 Result - Decoded Values

35=x|268=3 (message header)

279=0|269=2|270=9462.50|271=5|48=800123|22=8 (trade)

279=0|269=0|270=9462.00|271=175|1023=1|48=800123|22=8|346=15 (new bid 1)

279=0|269=0|270=9461.50|271=133|1023=2|48=800123|22=8|346=12 (new bid 2)

3.5 Sample Template

Line #	Template Syntax	Use and Description
2	<template name="**MDIncRefresh**" id="**35**">	Provides the template name (MDIncRefresh) and template identifier (35).
3	<typeRef name="**MDIncRefresh**" />	Specifies the type reference name that allows the template to be referenced in other templates.
4	<string name="**ApplVerID**" id="**1128**"> <constant value="**FIX.5.0**" /> </string>	Field instruction for ApplVerID defined as a string with an identifier of 1128 corresponding to the FIX tag number. ApplVerID has a constant field operator with a value of FIX.5.0 indicating the FIX version.
5	<string name="**MessageType**" id="**35**"> <constant value="**X**" /> </string>	Field instruction for MessageType defined as a string with identifier = 35 corresponding to the FIX tag number. MessageType has a constant field operator with a value of *X* which indicates the FIX message type—in this case Market Data Incremental Refresh.
6	<string name="**SenderCompID**" id="**49**"> <constant value="**CME**" /> </string>	Field instruction for SenderCompID defined as a string with identifier = 49 corresponding to the FIX tag number. SenderCompID has a constant field operator with a value of 'CME' which indicates the originator of the data.
7	<uInt32 name="**MsgSeqNum**" id="**34**"> <increment /> </uInt32>	Field instruction for MsgSeqNum defined as an unsigned integer with identifier = 34 corresponding to the FIX tag number. MsgSeqNum has an increment field operator.
8	<uInt64 name="**SendingTime**" id="**52**"> <delta /> </uInt64>	Field instruction for SendingTime defined as an unsigned integer and with identifier = 52 corresponding to the FIX tag number. SendingTime has a delta field operator.
9	<sequence name="**MDEntries**"> <length name="**NoMDEntries**" id="**268**"> <copy /> </length>	Sequence instruction demarks the beginning of the MDEntries repeating group. The sequence includes a length field called 'NoMDEntries' that specifies the number of repeating groups present in the message. NoMDEntries has a copy field operator
10	<int32 name="**MDEntryPx**" id="**270**"> <delta /> </int32>	Field instruction for MDEntryPx (first field instruction in repeating group) defined as a signed integer with identifier = 270 corresponding to the FIX tag number. MDEntryPx has a delta field operator.
11	<int32 name="**MDEntrySize**" id="**271**"> <delta /> </int32>	Field instruction for MDEntrySize defined as a signed integer with identifier = 271 corresponding to the FIX tag number. MDEntrySize has a delta field operator
12	<uInt32 name="**MDEntryTime**" id="**273**"> <delta /> </uInt32>	Field instruction for MDEntryTime which is defined as an unsigned integer with identifier = 273 corresponding to the FIX tag number. MDEntryTime has a delta field operator.

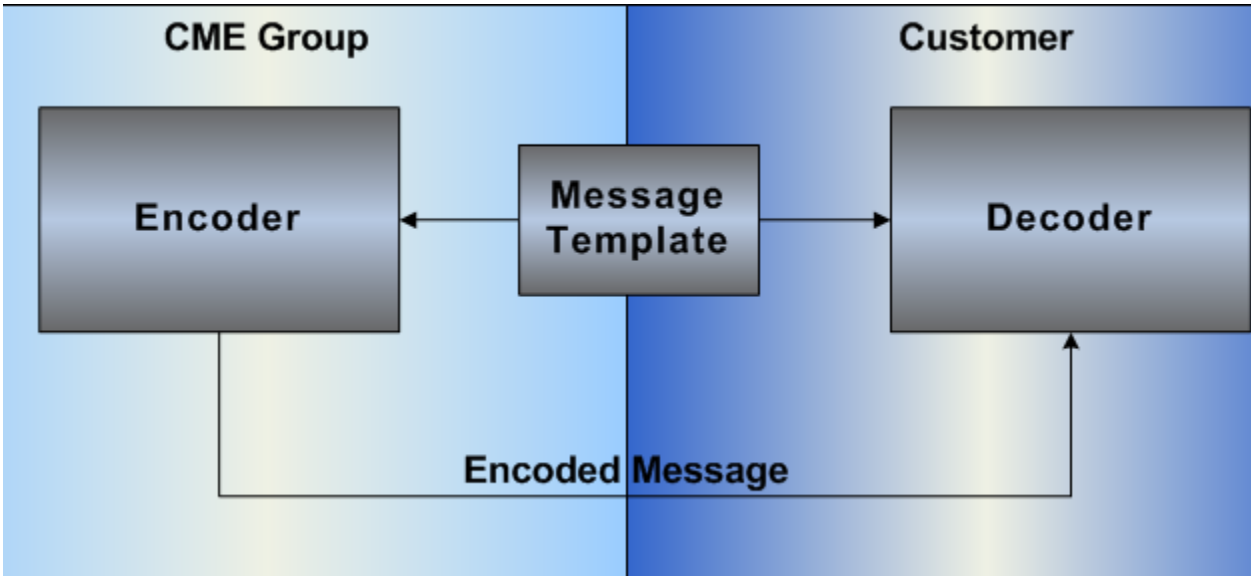
Line #	Template Syntax	Use and Description
13	<uInt32 name="**MDPriceLevel**" id="**1023**" presence="**optional**"> <increment /> </uInt32>	Field instruction for MDPriceLevel defined as an unsigned integer with identifier = 1023 corresponding to the FIX tag number. MDPriceLevel has a delta field operator.
14	<uInt32 name="**MDUpdateAction**" id="**279**"> <copy value="**0**" /> </uInt32>	Field instruction for MDUpdateAction defined as an unsigned integer and identifier = 279 corresponding to the FIX tag number. MDUpdateAction has a copy field operator with a value of 0.
15	<string name="**MDEntryType**" id="**269**"> <default value="**1**" /> </string>	Field instruction for MDEntryType which is defined as a string and has an identifier of 269 which corresponds to the FIX tag number. MDEntryType has a default field operator with a value of 1.
16	<int32 name="**SecurityID**" id=”**48**”><delta/></int32>	Field instruction for SecurityID defined as an unsigned integer and has an identifier = 48 corresponding to the FIX tag number. SecurityID has a delta field operator.
16	<uint32 name="**SecurityID**" idSource id=”**22**”><constant value="**8**"></uint32>	Field instruction for SecurityIDSource defined as an unsigned integer and has an identifier = 22 corresponding to the FIX tag number. SecurityDesc has a constant field operator.
17	<int32 name="**NumberOfOrders**" id="**346**" presence="**optional**"> <delta /> </int32>	Field instruction for NumberOfOrders defined as a signed integer and with identifier = 346 corresponding to the FIX tag number. NumberOfOrders has a delta field operator.
18	<string name="**QuoteCondition**" id="**276**" presence="**optional**"> <copy /> </string>	Field instruction for QuoteCondition defined as a string with identifier = 276 correspondsing to the FIX tag number. QuoteCondition has a copy field operator.
19	<string name="**TickDirection**" id="**274**" presence="**optional**"> <default /> </string>	Field instruction for TickDirection is defined as a string with identifier = 274 corresponding to the FIX tag number. TickDirection has a default field operator.
20	<uint32 name="**NetChgPrevDay**" id="**451**" presence="**optional**"> <default /> </uint32>	Field instruction for NetChgPrevDay defined as an unsigned integer with identifier = 451 corresponding to the FIX tag number. NetChgPrevDay has a default field operator.
21	<string name="**TradeCondition**" id="**277**" presence="**optional**"> <default /> </string>	Field instruction for TradeCondition defined as a string with identifier = 277 corresponding to the FIX tag number. TradeCondition has a default field operator.
22	<int32 name="**TradeVolume**" id="**1020**" presence="**optional**"> <default /> </int32>	Field instruction for TradeVolume defined as a signed integer with identifier = 1020 corresponding to the FIX tag number. TradeVolume has a default field operator.

Line #	Template Syntax	Use and Description
23	<string name="**TradingSessID**" id="**336**"> <default value="**1**" /> </string>	Field instruction for TradingSessID defined as a string with identifier = 336 corresponding to the FIX tag number. TradingSessID has a default field operator. Trading-SessID is the last field in the Sequence

4. Template Overview

CME Group provides a single xml file that contains a collection of templates and is shared across all market data channels between the encoder (CME Group side) and decoder (customer side). Each template has a unique template ID that describes the format of an encoded message. A template ID is carried in every encoded message to provide a reference to the correct template for decoding purposes.

A template selection process should be carried out on the customer side. The decoder should dynamically identify and retrieve the correct template indicated in the encoded message, then use that template to decode the message.



The template ID is an unsigned integer carried as the first data field following the first Pmap of every message allowing the decoding system to apply the correct template to the message upon receiving it.

Example: Template ID

```
2   <template name="MDIncRefresh" id="35" xmlns="http://www.fixprotocol.org/ns/fast/td/1.1">
3     <typeRef name="MDIncRefresh"/>
```

CME Group will not send the same template ID more than once in a UDP packet.

4.1 XML Template Example

A template consists of Field Instructions that define the fields contained in the message. Field Instructions specify the field name, tag number, data type, field operator, and presence attribute that indicates if a field is optional or mandatory.

A sample market data template is shown below. The syntax is standard XML and can be parsed using a variety of open source tools. Valid template syntax is determined by the FAST Template Schema which is available in the FAST v1.1 specification.

- The real-time feed templates are based on the Market Data Incremental Refresh message type.

The information contained in a template is passed to the client FAST decoder at run-time such that the decoder recognizes how each field is encoded in terms of data type representation (Transfer) and data redundancy removal (Field).

The template is constructed of several sections including Template Identification, Header, Body and Sequence. Template Identification provides the template name and identifier. The Header includes FIX header fields such as ApplVerID (tag 1128), MsgType (tag 35), and SendingTime (tag 52). The Body provides information common across all repeating groups. Sequence represents a repeating group with a corresponding length field and a set of repeating group fields which carry the detailed entry information.

```
<?xml version="1.0" encoding="UTF-8"?>
<template name="MDIncRefresh" id="30" xmlns="http://www.cme.com/mdp/fast-templates" >
  <typeRef name="MDIncRefresh"/>
  <string name="ApplVerID" id="1128"> <constant value="7"/> </string>
  <string name="MessageType" id="35"> <constant value="X"/> </string>
  <string name="SenderCompID" id="49"> <constant value="CME"/> </string>
  <uInt32 name="MsgSeqNum" id="34"> </uInt32>
  <uInt64 name="SendingTime" id="52"> <delta/> </uInt64>
  <uInt32 name="TradeDate" id="75"><copy/> </uInt32>
  <sequence name="MDEntries"><length name="NoMDEntries" id="268"></length>
    <decimal name="MDEntryPx" id="270">
      <exponent><copy value="-2"/></exponent>
      <mantissa><delta/></mantissa> </decimal>
    <int32 name="MDEntrySize" id="271"> <delta/> </int32>
    <uInt32 name="MDEntryTime" id="273"> <delta/> </uInt32>
    <uInt32 name="MDPriceLevel" id="1023" presence="optional"> <increment/> </uInt32>
    <uInt32 name="MDUpdateAction" id="279"> <copy value="1"/> </uInt32>
    <string name="MDEntryType" id="269"> <copy value="0"/> </string>
    <uInt32 name="SecurityID" id="48"> <copy/> </uInt32>
    <uInt32 name="SecurityIDSource" id="22"> <constant value="8"/> </uInt32>
    <int32 name="NumberOfOrders" id="346" presence="optional"> <delta/> </int32>
    <string name="QuoteCondition" id="276" presence="optional"> <copy value="K"/> </string>
    <string name="TickDirection" id="274" presence="optional"> <default/> </string>
    <int32 name="NetChgPrevDay" id="451" presence="optional"> <default/> </int32>
    <string name="TradeCondition" id="277" presence="optional"> <default/> </string>
    <int32 name="TradeVolume" id="1020" presence="optional"> <default/> </int32>
    <string name="TradingSessID" id="336"><default value="1"/> </string>
  </sequence>
</template>
```

4.2 Template Implementation Considerations

The following items should be considered before implementing template functionality:

- Client systems should use the defined sizes and type for each tag in the FIX Message Specifications as a guide for storing data. **Do not use the template to define this.**
- Any change to the template will result in an update to the template ID.
- The following template changes should be handled by the client without any changes to their decoder.

Note: **Customers will be notified prior to a template change via a Globex Notice and a Market Data Notice.**

Template Change	Decoder Impact	Template Release Plan	Additional Information
New Tags (CME Group originated)	none	A template that contains a new tag (CME Group originated) will be released in the New Release Certification environment approximately 4 weeks prior to Production environment (and Certification environment) roll-out.	A new tag may be defined as a result of a change in the business logic. This change should not require a modification to the client system decoder, however, there may be changes required to the clients back-end systems if they choose to implement this change.
New Tags (FIX originated)	none	A template that contains a new tag (FIX originated) will be released in the New Release environment approximately 4 weeks prior to Production environment (and Certification environment) roll-out.	A new tag may be defined as a result of a change in the business logic. This change should not require a modification to the client system decoder, however, there may be changes required to the clients back-end systems if they choose to implement this change.
Removed Tags	none	A template in which a tag is no longer available will be released in the New Release environment approximately 4 weeks prior to Production environment (and Certification environment) roll-out.	A tag may be removed, for example, if it is no longer needed.
Modified Tags	none	A template that contains a modified tag will be released in the New Release environment approximately 4 weeks prior to Production environment (and Certification environment) roll-out.	A modification to an existing tag may be defined as a result of a change in the business logic. This tag modification should not require a modification to the client system decoder, however, there may be changes required to the clients back-end systems if they choose to implement this change.

Template Change	Decoder Impact	Template Release Plan	Additional Information
Rename or Reorder Tags	none	A template that contains a renamed or reordered tag will be released in the New Release environment approximately 2 weeks prior to Production environment (and Certification environment) roll-out.	A renamed or reordered tag may be defined to increased efficiency. This tag modification should not require a modification to the client system decoder, however, there may be changes required to the clients back-end systems if they choose to implement this change.
New Message Types	none	A template that contains a new message type will be released in the New Release environment approximately 4 weeks prior to Production environment roll-out.	A new message type may be defined as a result of enhanced CME Globex functionality, or a change in business logic. This change should not require a modification to the client system decoder, however, there may be changes required to the clients back-end systems if they choose to implement this change.
Modified Operators	none	A template that contains a modified operator will be released in the New Release environment approximately 2 weeks prior to Production environment roll-out.	
Modified Data Types	none	A template that contains a modified data type will be released in the New Release environment approximately 2 weeks prior to Production environment roll-out.	

4.3 Template Distribution

The current template for each environment is available for download from the CME Group ftp site. Refer to “Services - Template Dissemination, Market Data Configuration, and Security Definition Flat File” on Page 9 for an outline of the process.

Note: **CME strongly recommends that you download the current templates file every Sunday prior to market open.**

Historical templates.xml files will be maintained on the ftp site and stored in a directory indicating the dates they were effective. The historical templates will be moved to the “Archive” directory for the corresponding environment.

4.4 Template Modification

A template archiving process is available. From the template ftp site (ftp.cmegroup.com), the ‘Templates’ directory for the corresponding environment (Production, New Release, and Certification) contains two sub-directories:

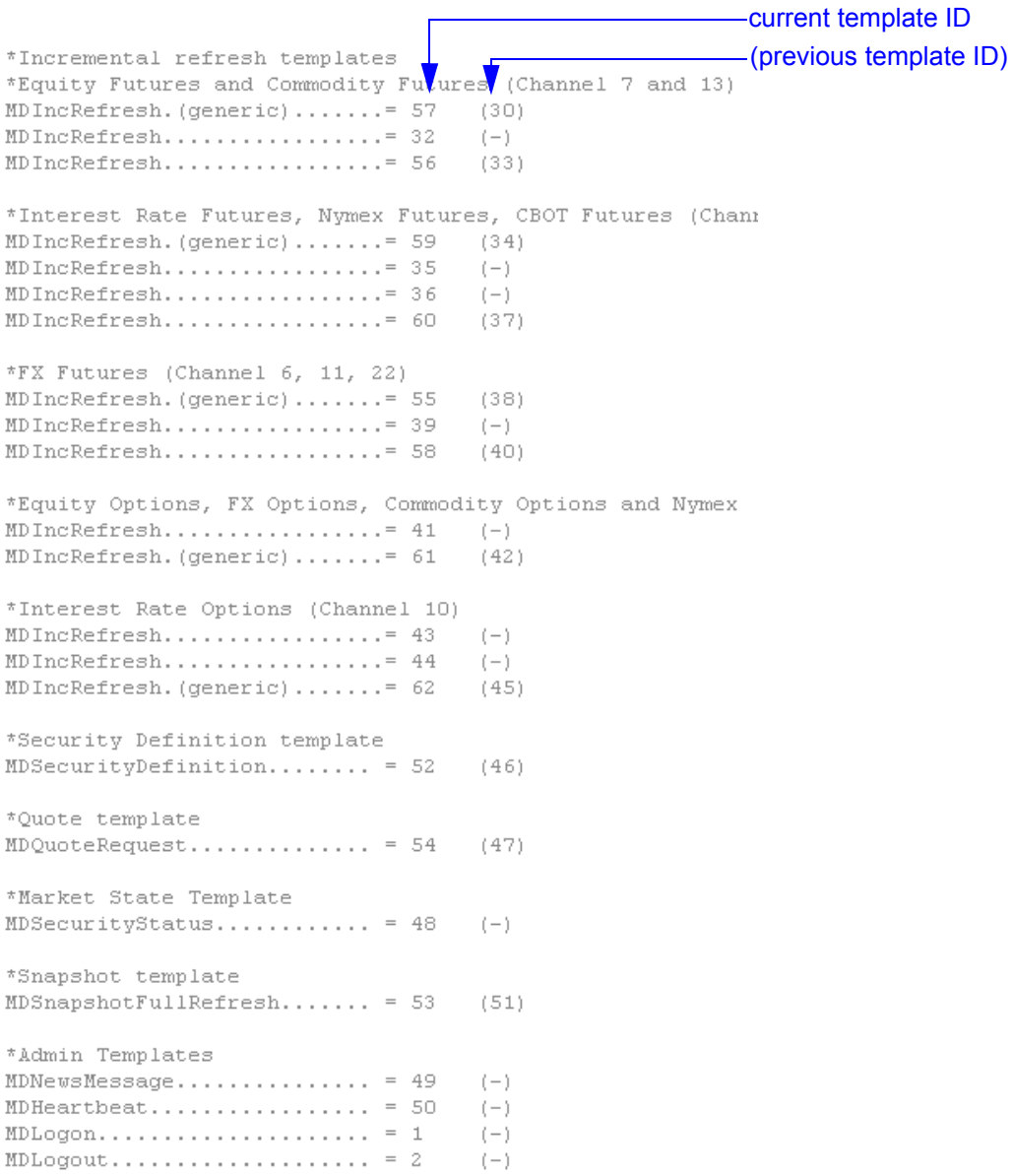
- **Active** - contains the current templates.xml file.
- **Archive** - contains all retired versions of the templates.xml file.

When modifying the templates.xml file, CME Group makes a copy of the active version, adds the next available template identifier, then makes the necessary changes in the templates.xml file and replaces it in the Active directory. Two versions of the template remain in the templates.xml file, with the previous template listed under the new template. When a template is changed for the third time, the oldest version of the template is moved to the templates.xml file in the Archive directory.

Example:

If template 30 previously replaced template 28 and now a new template 57 is being generated to replace template 30, then template 28 is removed from the templates.xml Active directory and added to the Archive directory. Templates 57 and 30 remain in the Active directory.

The template.xml file in the Active directory lists both the current and the previous template IDs, as displayed in the following example:



The following example displays the current template name “57” with the previous template name “30” listed below it. A note below the previous template indicates the current template ID.

```
  -->
+ <template name="MDIncRefresh_57" id="57" dictionary="57" xmlns="http://www.fixprotocol.org/ns/fast/td/1.1">
- <template name="MDIncRefresh_30" id="30" dictionary="30" xmlns="http://www.fixprotocol.org/ns/fast/td/1.1">
    <!-- desc="DEPRECATED. USE VERSION 57"  -->
  - <string name="ApplVerID" id="1128">
      <constant value="8" />
    </string>
```

4.5 Template Versioning

The **templates.xml** file is versioned each time an update is made. You can determine whether the client system has the latest templates or if the new **templates.xml** file needs to be downloaded and applied.

The version number changes whenever there is a change to the template, such as an:

- addition/removal/modification of a FIX tag
- addition/removal/modification of a FIX tag value
- addition of a new template

5. Incremental Book Management

5.1 Overview

CME Group uses an electronic market data format based on the FIX standard which provides a sound model for reducing the overall content of data transmitted through an incremental book management approach. FIX provides a flexible protocol for market data messaging which is well suited for transmitting electronic books from the CME Globex trading platform to customers. Incremental book management provides significant efficiency and flexibility across the entire market data infrastructure. The use of incremental FIX market data messaging in combination with FAST compression produces a highly optimized feed which results in bandwidth savings as well as latency reductions.

CME Group represents markets in executable orders and quotes using a central limit order book. This order book is constantly changing as market events cause orders to be added, modified, and cancelled. Updates are then sent out over a market data stream so client systems can then construct a copy of the book in order to track prices in the market and submit appropriately priced orders.

Note: **Multiple data blocks may be sent in the same message. To determine the number of data blocks you will receive in the message, refer to tag 268-NoMDEntries. Within the message, data blocks may be for different instruments or entry types (book update, statistics, or trades).**

5.2 FIX Message Structures

The Market Data Incremental Refresh (tag 35-MsgType = X) message is used to apply instructions to a book. These instructions are incremental and update applicable parts of the book as necessary, as opposed to refreshing the entire book each time there is an update. This message is used to maintain the aggregate order book for CME Group products. This message, on a real-time basis, is also used to send statistics.

The FIX Market Data message has a FIX header followed by a number of data blocks. Each data block represents a single instruction such as a book update or trade. This section refers to the items as data blocks when describing how the book is maintained. A single FIX Market Data message can contain many data blocks, such as New bid/ask, Change bid/ask, Delete bid/ask, and Overlay bid/ask, across multiple instruments.

5.2.1 Common Book Update Tags

Common Tags

Tag	Field Name	Description
279	MDUpdateAction	Type of Market Data update action.
269	MDEntryType	Type of Market Data entry.
83	RptSeq	Sequence number per Instrument update.
276	QuoteCondition	Space-delimited list of conditions describing a quote.
48	SecurityID	Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1023	MDPriceLevel	Position in the book.
273	MDEntryTime	Time of Market Data Entry.
270	MDEntryPx	Price of the Market Data Entry.
271	MDEntrySize	Quantity or volume represented by the Market Data Entry.
346	NumberOfOrders	Number of orders in the market.
336	TradingSessionID	Identifier for Trading Session.
1070	MDQuoteType	Identifies the type of quote.

5.3 Central Limit Order Book

The central limit order book is available over a market data feed that provides an aggregate book with a pre-determined number of price levels. The term “aggregate book” is used to refer to the case where orders are summarized at each price level. The quantity associated with that price level is the aggregation of all individual order quantities for that price level. CME Group offers the following:

- multiple-depth book
  - 3-deep book
  - 5-deep book
  - 10-deep book
- top-of-book
- 2-deep book for implied prices futures

Note: **Client systems must determine the book-depth for an instrument from tag 264-MarketDepth in the Security Definition (tag 35-MsgType=d) message.**

Note: **Client systems must be able to process all valid values in tag 279-MDUpdateAction.**

It is possible to deliver market data for instrument order books with different book depths on a single channel, for better bandwidth efficiency. For example, on Channel 7, Instrument A can have a 5-deep order book, while Instrument B has a 10-deep order book. Tag 264-MarketDepth in the Security Definition (Tag 35-MsgType=d) message must be referenced to obtain the book depth of an instrument.

Example - Multiple Order Book Depths

Market Data Channel	Instrument	Book Depth
Channel 7	Instrument A	5-Deep
Channel 7	Instrument B	10-Deep

5.4 Book Management Mechanics - Multiple-Depth Book

This section applies to CME Group products that have a multiple-depth book. Client systems must determine the book-depth for an instrument from tag 264-MarketDepth in the Security Definition (tag 35-MsgType=d) message.

The examples in this section illustrate the mechanics of a 5-deep book. All books that are **not top-of-book** will use the same mechanics. For example, a 3-deep book will use the same mechanics as described in this section. For top-of-book only, refer to “Book Management Mechanics - Top-of-Book” on Page 56.

5.4.1 Overview

CME Group provides a multiple-depth book for most products. Client systems must determine the book-depth for an instrument from tag 264-MarketDepth in the Security Definition (tag 35-MsgType=d) message. The aggregate book reports summarized order quantities and order counts at a given price level. The depth represents the number of price levels that are supported in the feed. The view can be represented as a number of rows in a table for each of the bid and ask sides. On each side, there are a number of rows showing the quantity available at a number of price levels. An aggregate depth book is sequenced by price, descending for bid and ascending for ask.

CME Group provides the best bid and ask in the market for each contract. Trade details and instrument status are provided in separate data blocks. The following table illustrates the types of information that can be displayed:

5-Deep - Best Bid/Ask

Bid			Ask		
Order Count	**Quantity**	**Price**	**Price**	**Quantity**	**Order Count**
1	100	9427.50	9428.00	40	2
19	500	9427.00	9428.50	600	35
34	750	9426.50	9429.00	850	55
25	400	9426.00	9429.50	350	21
14	300	9425.50	9430.00	150	12

CME Group maintains the Aggregate Depth view with the following data blocks:

- **Add** - to create/insert a new price at a specified price level (tag 279 MDUpdateAction=0)
- **Change** - change quantity for a price at a specified price level (tag 279 MDUpdateAction=1)
- **Delete** - remove a price at a specified price level (tag 279 MDUpdateAction=2)

An Aggregate book is built from a series of data blocks which indicate whether an entry is to be inserted (Add), changed (Change), or removed (Delete). All data blocks are issued for a specified entry type (tag 269), price (tag 270), and price level (tag 1023). The incremental instruction approach assumes the use of the Market Data Incremental Refresh (tag 35-MsgType =X) message. The Bid and Ask sides are updated independently with separate data blocks. The practice of sending separate data blocks provides efficiencies by allowing only the bid or ask to be sent, based on which side has changed, rather than both sides.

CME Globex sends an add data block if there is a new price level. Client systems should then shift price levels down, and delete any price levels past the defined depth of the book as indicated in tag 264-MarketDepth in the Security Definition (tag 35-MsgType=d) Message.

CME Globex sends a delete data block to remove a price level in the book. Client systems should shift prices below the data block up to the price level vacated by the deleted price level. If available, an add data block will be sent to fill in the last price level.

The change data block is sent to update characteristics of a price level without changing the price itself, or impacting any other prices on the book. The change data block is sent to update the order count and / or quantity for a price level. The change data block is not sent when the price changes at a given price level.

In general, if a trade occurs, CME Group will send a delete or change data block to update the book. The trade data block itself is not used to update the order book.

Note: **The last best price data block will appear similar to a book update data block. To identify that the data block is a last best price data block (and NOT a book update), verify that tag 276 - QuoteCondition contains a value of C (Exchange Best).**

5.4.1.1 Basic Book Update Data Block

A FIX message sent to update the top of book will update one side only. The tags normally sent for a book update data block are:

- tag 279-MDUpdateAction
- tag 269-MDEntryType
- tag 83-RptSeq
- tag 1023-MDPriceLevel
- tag 273-MDEntryTime
- tag 271-MDEntrySize
- tag 270-MDEntryPx
- tag 346-NumberOfOrders
- tag 48-SecurityID
- tag 22-SecurityIDSource

- tag 336-TradingSessionID

5.4.2 Examples

The following examples show how a book is built and updated. The book generally consists of a set of bid prices and ask prices, in which bids are descending and asks are ascending. The quantity and order count is provided at each price level.

The following table illustrates an initial market book.

5-Deep Book

Bid			Ask		
Order Count	**Quantity**	**Price**	**Price**	**Quantity**	**Order Count**
1	100	9427.50	9428.00	40	2
19	500	9427.00	9428.50	600	35
34	750	9426.50	9429.00	850	55
25	400	9426.00	9429.50	350	21
14	300	9425.50	9430.00	150	12

5.4.2.1 Quantity on Bid Side Modified

The quantity of an order can be modified. The book will show an update to the size displayed:

Bid 90 @ 9427.50

Book Update Instruction

Book Update - Data Block

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	1	1 = change. Type of Market Data update action.
269	MDEntryType	0	0 = bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
1023	MDPriceLevel	1	Price level 1. Position in the book.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	90	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9427.50	Price of the Market Data Entry.
346	NumberOfOrders		Number of orders in the market.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	8 = CME. Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

Book Update - Data Block

Tag Number	Tag Name	Value	Description
336	TradingSessionID		Identifier for Trading Session.

Updated Book

Bid			Ask		
Order Count	**Quantity**	**Price**	**Price**	**Quantity**	**Order Count**
1	90	9427.50	9428.00	40	2
19	500	9427.00	9428.50	600	35
34	750	9426.50	9429.00	850	55
25	400	9426.00	9429.50	350	21
14	300	9425.50	9430.00	150	12

5.4.2.2 Entire Order Canceled and New Order Entered

An entire order will be canceled, which removes one price level from the book, and a new order data block will be sent to fill the open price level 5. The book will show the removal of price level 1, followed by an addition to price level 5. A delete data block will be sent for trades.

Note: **Client systems should shift prices below the data block up to the price level vacated by the deleted price level. If all levels in the book are full, an add data block will be sent to fill in the last price level.**

Cancel bid 90 @ 9427.50

Book Update Instruction

Book Update - Data Block 1

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	2	2 = delete. Type of Market Data update action.
269	MDEntryType	0	0 = bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
1023	MDPriceLevel	1	Price level 3. Position in the book.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	0	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	0	Price of the Market Data Entry.
346	NumberOfOrders		Number of orders in the market.

Book Update - Data Block 1

Tag Number	Tag Name	Value	Description
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource		Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
336	TradingSessionID		Identifier for Trading Session.

Add bid 400 @ 9425.00

Book Update Instruction

Book Update - Data Block 2

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = add. Type of Market Data update action.
269	MDEntryType	0	0 = bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
1023	MDPriceLevel	5	Price level 5. Position in the book.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	400	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9425.00	Price of the Market Data Entry.
346	NumberOfOrders	1	Number of orders in the market.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource		Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
336	TradingSessionID		Identifier for Trading Session.

Updated Book

Bid			Ask		
Order Count	**Quantity**	**Price**	**Price**	**Quantity**	**Order Count**
19	500	9427.00	9428.00	40	2
34	750	9426.50	9428.50	600	35
25	400	9426.00	9429.00	850	55

Updated Book

Bid			Ask		
14	300	9425.50	9429.50	350	21
1	400	9425.00	9430.00	150	12

5.4.2.3 New Order Entered at Same Price

A new order can be entered to a book at the same price level. The book will show an update:

New bid order for 3 @ 9427.00

Book Update Instruction

Book Update - Data Block

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	1	1 = change. Type of Market Data update action.
269	MDEntryType	0	0 = bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
1023	MDPriceLevel	1	Price level 1. Position in the book.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	503	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9427.00	Price of the Market Data Entry.
346	NumberOfOrders	20	Number of orders in the market.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource		Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
336	TradingSessionID		Identifier for Trading Session.

Updated Book

Bid			Ask		
Order Count	**Quantity**	**Price**	**Price**	**Quantity**	**Order Count**
20	503	9427.00	9428.00	40	2
34	750	9426.50	9428.50	600	35
25	400	9426.00	9429.00	850	55
14	300	9425.50	9429.50	350	21

Updated Book

Bid			Ask		
1	400	9425.00	9430.00	150	12

5.4.2.4 New Best Price Entered

A new order can be entered to an existing book as a new best price. The data block indicates that a new order should be inserted at price level 1 (new best price). As a result, all orders on the book should be shifted down accordingly.

New bid order 200 @ 9427.50

Book Update Instruction

Book Update - Data Block

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data update action.
269	MDEntryType	0	0 = bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
1023	MDPriceLevel	1	Price level 1. Position in the book.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	200	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9427.50	Price of the Market Data Entry.
346	NumberOfOrders	1	Number of orders in the market.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource		Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
336	TradingSessionID		Identifier for Trading Session.

Updated Book

Bid			Ask		
Order Count	**Quantity**	**Price**	**Price**	**Quantity**	**Order Count**
1	200	9427.50	9428.00	40	2
20	503	9427.00	9428.50	600	35
34	750	9426.50	9429.00	850	55
25	400	9426.00	9429.50	350	21

Updated Book

Bid			Ask		
14	300	9425.50	9430.00	150	12

5.5 Book Management Mechanics - Implied Book

CME Group provides a 2-deep best bid and ask in the market for each implied prices futures contract. **Implied book updates are denoted by the presence of tag 276-QuoteCondition = K (Implied).** Trade details and instrument status are provided in separate data blocks. The following table illustrates the types of information that can be displayed:

2-Deep Book

Bid		Ask	
Quantity	**Price**	**Price**	**Quantity**
100	9427.50	9428.00	40
200	9427.00	9428.50	100

CME Group maintains the Aggregate Depth view with the following data blocks:

- **Add** - to create/insert a new price at a specified price level (tag 279-MDUpdateAction=0)
- **Change** - change quantity of a price at a specified price level (tag 279-MDUpdateAction=1)
- **Delete** - remove a price at a specified price level (tag 279-MDUpdateAction=2)

An Aggregate book is built from a series of data blocks which indicate whether an entry is to be inserted (Add), changed (Change), or removed (Delete). All data blocks are issued for a specified entry type (tag 269-MDEntryType), price (tag 270-MDEntryPx), price level (tag 1023-MDPriceLevel), and quantity (tag 271-MDEntrySize). The incremental instruction approach assumes the use of the Market Data Incremental Refresh message (tag 35-MsgType=X). The Bid and Ask sides are updated independently with separate data blocks. The practice of sending separate data blocks provides efficiencies by allowing only the bid or ask to be sent, based on which side has changed, rather than both sides.

CME Globex sends an add data block if there is a new price level. Client systems should then shift price levels down, and delete any price levels past 2-deep.

CME Globex sends a delete data block to remove a price level in the book. Client systems should shift prices below the data block up to the price level vacated by the deleted price level. If all levels in the book are full, an add data block will be sent to fill in the last price level.

The change data block is sent to update characteristics of a price level without changing the price itself, or impacting any other prices on the book. The change data block is sent to update the order quantity for a price level. The change data block is not sent when the price changes at a given price level.

If a trade occurs, CME Group will send a delete or change data block to update the book. The trade data block itself is not used to update the order book.

5.5.1 Basic Book Update Data Block

A FIX message sent to update the top of book will update one side only. The tags normally sent for a book update data block are:

Note: **tag 346-NumberOfOrders is not included in this data block.**

- tag 279-MDUpdateAction
- tag 269-MDEntryType
- tag 83-RptSeq
- tag 276-QuoteCondition
- tag 1023-MDPriceLevel
- tag 273-MDEntryTime
- tag 271-MDEntrySize
- tag 270-MDEntryPx
- tag 48-SecurityID
- tag 22-SecurityIDSource
- tag 336-TradingSessionID

5.5.2 Examples

This example shows how a book is built and updated. The book generally consists of a set of bid prices and ask prices, in which bids are descending and asks are ascending. The quantity is provided at each price level. For implied prices futures, a two-deep book is available.

The following table illustrates an initial implied market book.

2-Deep Book

Bid		**Ask**	
Quantity	**Price**	**Price**	**Quantity**
100	9427.50	9428.00	40
200	9427.00	9428.50	100

5.5.2.1 Quantity on Bid Side Modified

The quantity of an order can be modified. The book will show an update to the size displayed:

Bid 90 @ 9427.50

Book Update Instruction

Book Update - Data Block

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	1	1 = Change. Type of Market Data update action.
269	MDEntryType	0	0 = Bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
276	QuoteCondition		Space-delimited list of conditions describing a quote.
1023	MDPriceLevel	1	Price Level 1. Position in the book.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	90	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9427.50	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource		Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
336	TradingSessionID		Identifier for Trading Session.

Updated Book

Bid		Ask	
Quantity	**Price**	**Price**	**Quantity**
90	9427.50	9428.00	40
200	9427.00	9428.50	100

5.5.2.2 Entire Order Canceled and New Order Entered

An entire order can be canceled, which removes one price level, and a new order data block will be sent to fill the open price level 2. The book will show the removal of the first price level, followed by an addition to the second price level.

Note: **Client systems should shift prices below the data block up to the price level vacated by the deleted price level. If all levels in the book are full, an add data block will be sent to fill in the last price level.**

Cancel bid 200 @ 9427.00

Book Update Instruction

Book Update - Data Block 1

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	2	2 = Delete. Type of Market Data update action.
269	MDEntryType	0	0 = Bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
276	QuoteCondition		Space-delimited list of conditions describing a quote.
1023	MDPriceLevel	2	Price Level 2. Position in the book.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	200	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9427.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource		Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
336	TradingSessionID		Identifier for Trading Session.

Add bid 80 @ 9426.50

Book Update Instruction

Book Update - Data Block 2

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = Add. Type of Market Data update action.
269	MDEntryType	0	0 = Bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
276	QuoteCondition		Space-delimited list of conditions describing a quote.
1023	MDPriceLevel	2	Price Level 2. Position in the book.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	80	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9426.50	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource		Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
336	TradingSessionID		Identifier for Trading Session.

Updated Book

Bid		Ask	
Quantity	**Price**	**Price**	**Quantity**
90	9427.50	9428.00	40
80	9426.50	9428.50	100

5.5.2.3 New Order Entered at Same Price

A new order can be entered to a book at the same price level. The book will show an update:

New bid order 3 @ 9427.50

Book Update Instruction

Book Update - Data Block 1

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	1	1 = Change. Type of Market Data update action.
269	MDEntryType	0	0 = Bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
276	QuoteCondition		Space-delimited list of conditions describing a quote.
1023	MDPriceLevel	1	Price Level 1. Position in the book.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	93	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9427.50	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource		Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
336	TradingSessionID		Identifier for Trading Session.

Updated Book

Bid		Ask	
Quantity	**Price**	**Price**	**Quantity**
93	9427.50	9428.00	40
80	9426.50	9428.50	100

5.6 Consolidating Implied and Multiple-Depth Books into a Single Book

The data block for a multiple-depth book reports modifications to prices, quantities, and order count, and the data block for an implied prices book reports modifications to prices and quantities. The multiple-depth book should be used in conjunction with implied prices book to create an accurate book for all contracts with implied functionality. To create a consolidated book, the multiple-depth book and the implied book must be built and managed separately, then consolidated to reflect the current state of the market.

Multiple-Depth Book

Bid			Ask		
Order Count	**Quantity**	**Price**	**Price**	**Quantity**	**Order Count**
19	500	9427.00	9428.00	40	2
34	750	9426.50	9428.50	600	35
25	400	9426.00	9429.00	850	55
14	300	9425.50	9429.50	350	21
10	200	9425.00	9430.50	150	12

Implied Book

Bid		Ask	
Quantity	**Price**	**Price**	**Quantity**
100	9427.50	9428.00	40
200	9427.00	9430.00	100

After building the multiple-depth book and implied books, the actual book can be built by merging the multiple-depth book and implied book tables.

Note: **Order count is not displayed in the combined book.**

Consolidated Book

Bid		Ask	
Quantity	**Price**	**Price**	**Quantity**
100	9427.50	9428.00	80
700	9427.00	9428.50	600
750	9426.50	9429.00	850
400	9426.00	9429.50	350
300	9425.50	9430.00	100

5.7 Book Management Mechanics - Top-of-Book

This section applies to CME Group products that are top-of-book. Client systems must determine the book-depth for an instrument from tag 264-MarketDepth in the Security Definition (tag 35-MsgType=d) message.

5.7.1 Overview

CME Group provides a multiple-depth book for several products. Client systems must determine the book-depth for an instrument from tag 264-MarketDepth in the Security Definition (tag 35-MsgType=d) Message.

CME Group provides the current best bid and ask (top-of-book) in the market for certain instruments. Client systems must determine the book-depth for an instrument from tag 264-MarketDepth in the Security Definition (tag 35-MsgType=d) Message. Trade details and instrument status are provided in separate data blocks. The following table illustrates the types of information that can be displayed:

1-deep Book - Best Bid/Ask

Bid			Ask		
Order Count	**Quantity**	**Price**	**Price**	**Quantity**	**Order Count**
1	100	9427.50	9428.00	40	2

CME Group maintains a top-of-book view with the following data block:

- **Overlay** - add, modify, or delete an entry in the book. (tag 279 MDUpdateAction=5)

Note: **When the best price changes, CME Globex sends a single overlay instruction for price level 1. Overlay is only used for instruments with order books that are one level deep.**

The incremental instruction approach assumes the use of the Market Data Incremental Refresh message (tag 35-MsgType =X). The Bid and Ask sides are updated independently with separate data blocks. The practice of sending separate data blocks provides efficiencies by allowing only the bid or ask to be sent, based on which side has changed, rather than both sides.

5.7.1.1 Basic Book Update Data Block

A FIX message sent to update a top-of-book updates one side only. The tags normally sent for a top-of-book data block are:

- tag 279-MDUpdateAction
- tag 269-MDEntryType
- tag 83-RptSeq
- tag 1023-MDPriceLevel
- tag 273-MDEntryTime
- tag 271-MDEntrySize
- tag 270-MDEntryPx

- tag 346-NumberOfOrders
- tag48-SecurityID
- tag22-SecurityIDSource
- tag 336-TradingSessionID

5.7.1.2 Indexing the Entry

Top-of-book entries are referenced using a composite index which consists of the Security Description, Tag 269-MDEntryType (bid/ask) and Tag 270-MDEntryPx. This set of fields acts as a composite key that allows the entry (bid or ask) to be accessed and subsequently, updated or deleted.

5.7.2 Examples

This example shows how a book is built and updated. The quantity and order count are provided.

The following table illustrates an initial market book.

Top-of-Book - Best Bid/Ask

Bid			Ask		
Order Count	**Quantity**	**Price**	**Price**	**Quantity**	**Order Count**
1	100	9427.50	9428.50	40	2

5.7.2.1 Quantity on Bid Side Modified

The quantity of an order can be modified. The book will show an update to the size displayed:

Bid 90 @ 9427.50

Book Update Instruction

Book Update - Data Block 1

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	5	5 = overlay. Type of Market Data update action.
269	MDEntryType	0	0 = bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
1023	MDPriceLevel	1	Price level 1. Position in the book.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	90	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9427.50	Price of the Market Data Entry.
346	NumberOfOrders	1	Number of orders in the market.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.

Book Update - Data Block 1

Tag Number	Tag Name	Value	Description
22	SecurityIDSource		Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
336	TradingSessionID		Identifier for Trading Session.

Top-of-Book - Best Bid/Ask

Bid			Ask		
Order Count	**Quantity**	**Price**	**Price**	**Quantity**	**Order Count**
1	90	9427.50	9428.50	40	2

5.7.2.2 Entire Order Canceled

An entire order can be canceled, which removes the top of book. The book will show an update:

Cancel bid 90 @ 9427.50

Book Update Instruction

Book Update - Data Block 1

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	5	5 = overlay. Type of Market Data update action.
269	MDEntryType	0	0 =bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
1023	MDPriceLevel	1	Price level 1. Position in the book.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	0	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx		Price of the Market Data Entry.
346	NumberOfOrders	0	Number of orders in the market.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource		Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
336	TradingSessionID		Identifier for Trading Session.

Top-of-Book - Best Bid/Ask

Bid			Ask		
Order Count	**Quantity**	**Price**	**Price**	**Quantity**	**Order Count**
			9428.50	40	2

5.7.2.3 New Order Entered

A new order can be entered to a book. The book will show an update:

New bid order 10 @ 9428.00

Book Update Instruction

Book Update - Data Block 1

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	5	5 = overlay. Type of Market Data update action.
269	MDEntryType	0	0 =bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
1023	MDPriceLevel	1	Price level 1. Position in the book.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	10	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9428.00	Price of the Market Data Entry.
346	NumberOfOrders	1	Number of orders in the market.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource		Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
336	TradingSessionID		Identifier for Trading Session.

Top-of-Book - Best Bid/Ask

Bid			Ask		
Order Count	**Quantity**	**Price**	**Price**	**Quantity**	**Order Count**
1	10	9428.00	9428.50	40	2

5.7.2.4 New Order Entered at Same Price

A new order can be entered to a book at the same price level. The book will show an update:

New bid order 3 @ 9428.00

Book Update Instruction

Book Update - Data Block 1

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	5	5 = overlay. Type of Market Data update action.
269	MDEntryType	0	0 = bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
1023	MDPriceLevel	1	Price level 1. Position in the book.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	13	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9428.00	Price of the Market Data Entry.
346	NumberOfOrders	2	Number of orders in the market.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource		Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
336	TradingSessionID		Identifier for Trading Session.

Top-of-Book - Best Bid/Ask

Bid			Ask		
Order Count	**Quantity**	**Price**	**Price**	**Quantity**	**Order Count**
2	13	9428.00	9428.50	40	2

6. Real Time Statistics (Market Behavior Events)

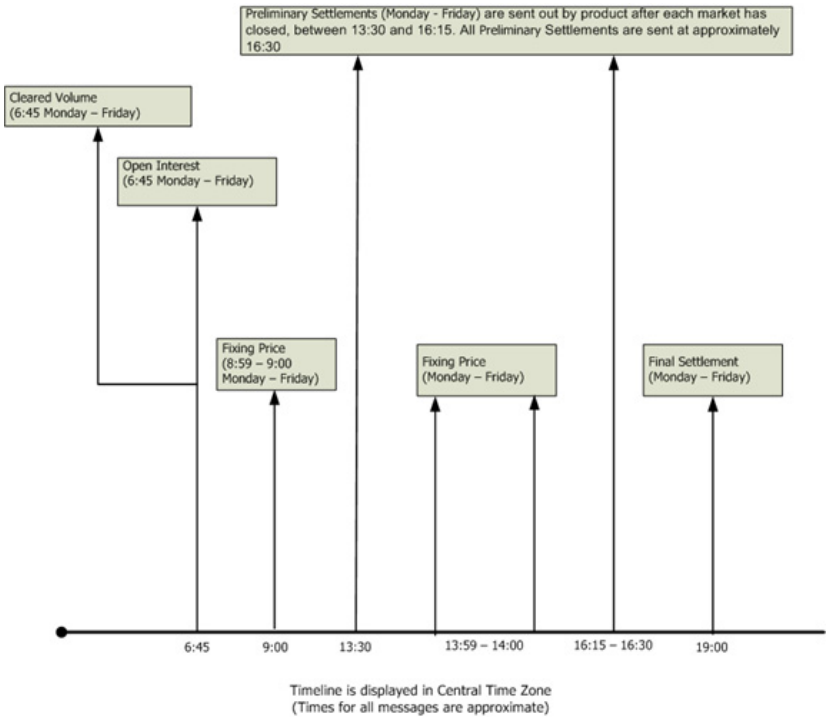
There are a number of statistics (market data events) which are related to changes in a book but are not used to update the book. The following artifacts fit this category: last best price, trade, high/low trade price, best high bid, and best low ask, price, and pre-opening statistics. These events describe the behavior of the market and allow a user to know when the market is moving in a certain direction and provide historical information on how the market has performed.

Note: **Multiple data blocks may be sent in the same message. To determine the number of data blocks you will receive in the message, refer to tag 268-NoMDEntries. Within the message, data blocks may be for different instruments or entry types (book update, statistics, or trades).**

6.1 CME Globex Message Timeline

The following timeline shows when certain messages are typically sent on a given trading day.

Note: **The following times are subject to change without notice.**



6.2 Pre-Opening Statistics

Pre-opening statistics are used to indicate simulated buy or sell, stop spike, and pre-opening prices. Simulated buy provides the theoretical best buy price that would be present after opening. Simulated sell provides the theoretical best sell price that would be present after opening.

Prior statistics are also available.

FIX Syntax for Prior - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data update action.
269	MDEntryType	M	M = prior. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
336	TradingSessionID		Identifier for Trading Session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.

FIX Syntax for Opening - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data update action.
269	MDEntryType	4	4 = opening price. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

FIX Syntax for Simulated Sell - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data update action.
269	MDEntryType	E	E = simulated sell. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
271	MDEntrySize		Quantity or volume represented by the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

FIX Syntax for Simulated Buy - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data update action.
269	MDEntryType	F	F = simulated buy. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
271	MDEntrySize		Quantity or volume represented by the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

6.3 Last Best Price

The Last Best Price data block indicates that a new best bid or new best ask price has occurred, and implicitly conveys events which affect the top of the book. **Last Best Price data blocks are denoted by the presence of tag 276-QuoteCondition = C (Exchange Best).**

In certain cases, such as an order submission immediately followed by an order cancellation, the top of the book can be bettered without resulting in a new top of book data block being sent. This is due to the transitory nature of the event. However, last best price is used to capture this information and convey it to the market.

FIX Syntax for Last Best Price (Bid or Ask) - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data update action.
269	MDEntryType	0 or 1	0 = bid and 1 = ask. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
276	QuoteCondition	C	C = exchange best. Space-delimited list of conditions describing a quote.

Note: **Tag 277-TradeCondition is not sent for this data block.**

6.4 Trade

The trade data block is sent when a trade occurs to provide volume and trade statistics. The following sections detail various trade types and their respective data blocks that demonstrate the use of Tag 277-TradeCondition.

Tag 277-TradeCondition is present and set to 1 when the reported trade price is an assigned price by CME Globex.

Tag 277-TradeCondition is not present when the reported trade price is not assigned by CME Globex.

When Tag 277-TradeCondition is not present or Tag 277-TradeCondition=E (opening trade), then the data represents the last trade.

Tag 269-MDEntryType=2 (trade) is not used to update the view of the book.

The trade data block may indicate whether the aggressor of the trade is on the buy side or the sell side. A trade data block may also be flagged as either the beginning or the end of a CME Globex event.

6.4.1 Example 1 - Two Outright Orders Trading

The following example illustrates the trade data block that is sent when a buy and sell match on GEZ2.

FIX Syntax for GEZ2 Outright Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.
5797	AggressorSide	1 or 2	1 = Buy and 2 = Sell. Indicates if the aggressor is on the buy or sell side of the order on the book that triggers a trade.
5799	MatchEventIndicator	1 or 2	1 = Beginning and 2 = End. Indicates the beginning or the end of a CME Globex event. If there is no value present, then the message is not at the beginning or the end of a CME Globex event. **Note:** Currently, only 1 is a valid value. 2 will become a valid value in a future release.

Note: **Tag 277-TradeCondition is not sent for this data block.**

6.4.2 Example 2 - Two Spread Orders Trading (Not Implied)

Several trade data blocks are sent for a spread. A trade may be executed in an off-market manner, and in this case the leg trade prices are CME-calculated and may not reflect the current market prices.

The following example illustrates the data blocks that are sent for a match event involving one order on ZCM8-ZCU8 against one order on ZCM8-ZCU8.

The resulting trade data blocks that will occur are:

- One trade data block for ZCM8ZCU8 (tag 277-TradeCondition is not sent)
- One trade data block for ZCM8 (tag 277-TradeCondition = 1)
- One trade data block for ZCU8 (tag 277-TradeCondition = 1)

FIX Syntax for ZCM8-ZCU8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	1360.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.
5797	AggressorSide	1 or 2	1 = Buy and 2 = Sell. Indicates if the aggressor is on the buy or sell side of the order on the book that triggers a trade.

FIX Syntax for ZCM8-ZCU8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
5799	MatchEventIndicator	1 or 2	1 = Beginning and 2 = End. Indicates the beginning or the end of a CME Globex event. If there is no value present, then the message is not at the beginning or the end of a CME Globex event. **Note:** Currently, only 1 is a valid value. 2 will become a valid value in a future release

Note: **Tag 277-TradeCondition is not sent for this data block.**

FIX Syntax ZCM8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	1360.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.
277	TradeCondition	1	Space-delimited list of conditions describing a trade. • 1 = Price calculated by CME Globex. **Note:** Tag 277- TradeCondition is not sent for a last best price data block.

FIX Syntax ZCM8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
5797	AggressorSide	1 or 2	1 = Buy and 2 = Sell. Indicates if the aggressor is on the buy or sell side of the order on the book that triggers a trade.

FIX Syntax ZCU8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	1360.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.
277	TradeCondition	1	Space-delimited list of conditions describing a trade. • 1 = Price calculated by CME Globex. **Note:** Tag 277 - TradeCondition is not sent for a last best price data block.
5797	AggressorSide	1 or 2	1 = Buy and 2 = Sell. Indicates if the aggressor is on the buy or sell side of the order on the book that triggers a trade.

6.4.3 Example 3 - Spread Order Trading Against Two Outright Orders (Implied)

The following example illustrates the data blocks that are sent for a match event involving one order on GEM8-GEU8 against one order on GEM8 and one order on GEU8.

The resulting trade data blocks that will occur are:

- One trade data block for GEM8-GEU8 (tag 277-TradeCondition is not sent)
- One trade data block for GEM8 (tag 277-TradeCondition is not sent)
- One trade data block for GEU8 (tag 277-TradeCondition is not sent)

FIX Syntax for GEM8-GEU8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.
5797	AggressorSide	1 or 2	1 = Buy and 2 = Sell. Indicates if the aggressor is on the buy or sell side of the order on the book that triggers a trade.

FIX Syntax for GEM8-GEU8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
5799	MatchEventIndicator	1 or 2	1 = Beginning and 2 = End. Indicates the beginning or the end of a CME Globex event. If there is no value present, then the message is not at the beginning or the end of a CME Globex event. **Note:** For this release, only 1 is a valid value. 2 will become a valid value in a future release

Note: **Tag 277-TradeCondition is not sent for this data block.**

FIX Syntax GEM8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.
5797	AggressorSide	1 or 2	1 = Buy and 2 = Sell. Indicates if the aggressor is on the buy or sell side of the order on the book that triggers a trade.

Note: **Tag 277-TradeCondition is not sent for this data block.**

FIX Syntax GEU8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.
5797	AggressorSide	1 or 2	1 = Buy and 2 = Sell. Indicates if the aggressor is on the buy or sell side of the order on the book that triggers a trade.

Note: **Tag 277-TradeCondition is not sent for this data block.**

6.4.4 Example 4 - Butterfly Order Trading Against a Calendar Order and Two Outright Orders (Implied)

The following example illustrates the data blocks that are sent for a match event involving one order on GE:BF M8-U8-Z8 against one order on GEM8-GEU8, one order on GEU8, and one order on GEZ8.

The resulting trade data blocks that will occur are:

- One trade data block for GE:BF M8-U8-Z8 (tag 277-TradeCondition is not sent)
- One trade data block for GEM8-GEU8 (tag 277-TradeCondition is not sent)

- One trade data block for GEM8 (tag 277-TradeCondition = 1)
- One trade data block for GEU8 (tag 277-TradeCondition = 1)
- One trade data block for GEU8 (tag 277-TradeCondition is not sent)
- One trade data block for GEZ8 (tag 277-TradeCondition is not sent)

FIX Syntax GE:BF M8-U8-Z8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.
5797	AggressorSide	1 or 2	1 = Buy and 2 = Sell. Indicates if the aggressor is on the buy or sell side of the order on the book that triggers a trade.
5799	MatchEventIndicator	1 or 2	1 = Beginning and 2 = End. Indicates the beginning or the end of a CME Globex event. If there is no value present, then the message is not at the beginning or the end of a CME Globex event. **Note:** Currently, only 1 is a valid value. 2 will become a valid value in a future release

Note: **Tag 277-TradeCondition is not sent for this data block.**

FIX Syntax GEM8-GEU8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.
5797	AggressorSide	1 or 2	1 = Buy and 2 = Sell. Indicates if the aggressor is on the buy or sell side of the order on the book that triggers a trade.

Note: **Tag 277-TradeCondition is not sent for this data block.**

FIX Syntax GEM8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.

FIX Syntax GEM8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.
277	TradeCondition	1	Space-delimited list of conditions describing a trade. • 1 = Price calculated by CME Globex. **Note:** Tag 277 - TradeCondition is not sent for a last best price data block.
5797	AggressorSide	1 or 2	1 = Buy and 2 = Sell. Indicates if the aggressor is on the buy or sell side of the order on the book that triggers a trade.

FIX Syntax for GEU8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.

FIX Syntax for GEU8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.
277	TradeCondition	1	Space-delimited list of conditions describing a trade. • 1 = Price calculated by CME Globex. **Note:** Tag 277 - TradeCondition is not sent for a last best price data block.
5797	AggressorSide	1 or 2	1 = Buy and 2 = Sell. Indicates if the aggressor is on the buy or sell side of the order on the book that triggers a trade.

FIX Syntax GEU8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.

FIX Syntax GEU8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.
5797	AggressorSide	1 or 2	1 = Buy and 2 = Sell. Indicates if the aggressor is on the buy or sell side of the order on the book that triggers a trade.

Note: **Tag 277-TradeCondition is not sent for this data block.**

FIX Syntax GEZ8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.
5797	AggressorSide	1 or 2	1 = Buy and 2 = Sell. Indicates if the aggressor is on the buy or sell side of the order on the book that triggers a trade.

Note: **Tag 277-TradeCondition is not sent for this data block.**

6.5 Session High and Low Trade Price

The High Trade Price data block is sent when a trade occurs which results in the highest trade price for the current session. Likewise, the Low Trade Price data block is sent when a trade occurs which results in the lowest trade price for the current session. High and low trade prices are helpful in tracking market trends. They also provide historical information for the current session regarding market behavior.

CME Group will provide high and low trade prices only as a response to the event that produces them.

FIX Syntax for Session High Trade Price - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data update action.
269	MDEntryType	7	7 = session high. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

FIX Syntax for Session Low Trade Price- Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data update action.
269	MDEntryType	8	8 = session low. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

6.6 Best High Bid and Best Low Ask

The best high bid and best low ask prices are used to indicate the highest bid and lowest ask prices of the session. These prices are useful in tracking market behavior. A best high bid and best low ask may be different from the high trade price described above if an arriving order is given a price better than the specified limit order. An order to buy at 9751 may trade at 9750 if a resting sell order is already on the book at 9750. In this case, the best high bid price would be higher than the high trade price.

FIX Syntax for Best High Bid - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data update action.
269	MDEntryType	N	N = session high bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize		Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

FIX Syntax for Best Low Ask - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data update action.
269	MDEntryType	O	O = session low ask. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize		Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.

FIX Syntax for Best Low Ask - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

6.7 Open Interest

Open Interest is sent using Market Data Incremental Refresh (tag 35-MsgType=X) message data blocks which contain the total number of contracts per instrument that are not yet offset or fulfilled for the previous trading day. They are sent before the start of the trading session. Tag 75-TradeDate will contain the date of the previous trading day. Tag 279-MDUpdateAction will be set to 0 = New and tag 269-MDEntryType will be set to C = Open Interest. Tag 271-MDEntrySize will contain the quantity or volume of the Open Interest.

Open Interest is updated throughout the trading week. Client systems can leverage the Security Definition (tag 35-MsgType=d) message to recover the last published open interest for an instrument. With this release, on Sunday start, the Security Definition (tag 35-MsgType=d) Message is published initially to the Instrument Definition feeds and the Market Data Incremental Refresh feeds without tag 5792-OpenInterestQty. Once open interest is sent in the Market Data Incremental Refresh (tag 35-MsgType=X) Message, the value of open interest from tag 271-MDEntrySize is placed on the Security Definition (tag 35-MsgType=d) Message in tag 5792-OpenInterestQty and maintained on the Instrument Definition feed only. As open interest updates are disseminated on the Market Data Incremental Refresh feed throughout the week, the value of tag 5792-OpenInterestQty is updated.

Note: **If open interest is 0, tag 5792-OpenInterestQty=0.**

6.7.1 Open Interest Data Block Sample Message

The following sample message contains the Open Interest data block.

1128=8|35=X|49=CME|34=6957050|52=20080124064553918|**75=20080124|**268=1|**279=0|**22=8|48=64288|83=127|**269=C|271=2000|**273=064553000|

Table 6.1. Market Data Incremental Refresh (tag 35-MsgType=X) Message

Tag	FIX Name	Format	Sample Values	Description
FIX Header - **1128=8|9=118|35=X|49=CME|34=6957050|52=20080124163153918|75=20080124|268=1|**				
279	MDUpdateAction	Char (1)	0	Type of Market Data update action. 0 = New
22	SecurityIDSource	String (1)	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48- SecurityID is specified.

Table 6.1. Market Data Incremental Refresh (tag 35-MsgType=X) Message

Tag	FIX Name	Format	Sample Values	Description
48	SecurityID	String (12)	64288	Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
83	RptSeq	Int (1)	127	Sequence number per Instrument update.
269	MDEntryType	Char (1)	C	Type of Market Data Entry. C = Open Interest
271	MDEntrySize	Qty(12)	2000	Quantity or volume represented by the Market Data Entry.
273	MDEntryTime	UTCTime Only (12)	064553000	Time of Market Data Entry. The Market Data entry time is 06:45:53.000 UTC time

6.8 Cleared Volume

Cleared Volume data blocks contain the number of contracts that have been through the clearing process for an active instrument for the previous trading day. Tag 75-TradeDate will contain the current trade date. Tag 279-MDUpdateAction will be set to 0 = New and tag 269-MDEntryType will be set to B = Trade Volume. Tag 271-MDEntrySize will contain the Cleared Volume.

Cleared Volume is updated throughout the trading week. Client systems can leverage the Security Definition (tag 35-MsgType=d) Message to recover the last published cleared volume for an instrument. On Sunday start, the Security Definition (tag 35-MsgType=d) Message is published initially to the Instrument Definition feeds and the Market Data Incremental Refresh feeds without tag 5791-ClearedVolume. Once Cleared Volume is sent in the Market Data Incremental Refresh (tag 35-MsgType=X) message, the value of Cleared Volume from tag 271-MDEntrySize is placed on the Security Definition (tag 35-MsgType=d) Message in tag 5791-ClearedVolume and maintained on the Instrument Definition feed. As cleared volume updates are disseminated on the Market Data Incremental Refresh feed throughout the week, the value of tag 5791-ClearedVolume is updated.

Note: **If cleared volume is 0, tag 5791-ClearedVolume=0.**

6.8.1 Cleared Volume Data Block Sample Message

The following sample message contains the Cleared Volume data block.

1128=8|35=X|49=CME|34=6957050|52=20080124163153918|**75=20080124**|268=1|**279=0|**22=8|48=64288|83=127|**269=B|271=2000|**273=163153000|

Table 6.2. Market Data Incremental Refresh (tag 35-MsgType=X) Message

Tag	FIX Name	Format	Sample Values	Description
FIX Header - **1128=8|35=X|49=CME|34=6957050|52=20080124163153918|75=20080124|268=1|**				
279	MDUpdateAction	Char (1)	0	Type of Market Data update action. 0 = New
22	SecurityIDSource	String (1)	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48- SecurityID is specified.
48	SecurityID	String (12)	64288	Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
83	RptSeq	Int (1)	127	Sequence number per Instrument update.
269	MDEntryType	Char (1)	B	Type of Market Data Entry. B = Trade Volume
271	MDEntrySize	Qty(12)	2000	Quantity or volume represented by the Market Data Entry.
273	MDEntryTime	UTCTime Only (12)	163153000	Time of Market Data Entry. The Market Data entry time is 16:31:53:000 UTC time

6.9 Fixing Price

Fixing Price data blocks contain the Fixing Price, a volume-weighted average price for the nearby futures contract of an Option, and the time the Fixing Price was calculated (tag 5790-FixingBracket). Tag 279-MDUpdateAction will be set to 0 = New and tag 269-MDEntryType will be set to W = Fixing Price.

6.9.1 Fixing Price Data Block Sample Message

The following sample message contains the Fixing Price data block.

1128=8|35=X|49=CME|34=6957050|52=20080124134553918|75=20080124|268=1|**279=0|**22=8|48=64288|83=127|**269=W|**270=2000|273=134553000|**5790=14:00**

Table 6.3. Market Data Incremental Refresh (tag 35-MsgType=X) Message

Tag	FIX Name	Format	Sample Values	Description
FIX Header - **1128=8|35=X|49=CME|34=6957050|52=20080124134553918|75=20080124|268=1|**				
279	MDUpdateAction	Char (1)	0	Type of Market Data update action. 0 = New
22	SecurityIDSource	String (1)	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48- SecurityID is specified.
48	SecurityID	String (12)	64288	Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
83	RptSeq	Int (1)	127	Sequence number per Instrument update.
269	MDEntryType	Char (1)	W	Type of Market Data Entry. W = Fixing Price
270	MDEntryPx	Price (20)	2000	Price of the Market Data Entry
273	MDEntryTime	UTCTime Only (12)	134553000	Time of Market Data Entry. The Market Data entry time is13:45:53.000 UTC time
5790	FixingBracket	Char (5)	14:00	Time that the Fixing Price was calculated.

6.10 Settlement Price

Settlement price is updated throughout the trading week. Client systems can leverage the Security Definition (tag 35-MsgType=d) message to recover the last published settlement price for an instrument. Settlement price updates are initially sent on the Market Data Incremental Refresh (tag 35-MsgType=X) message on the Market Data Incremental Refresh feed. When an update occurs to the settlement price, it is captured and placed on the Security Definition (tag 35-MsgType=d) message for each instrument on the Instrument Definition feed. As new settlement prices are published, the date (tag 5796-TradingReferenceDate) and price (tag 1150-TradingReferencePrice) are updated. Tag 731-SettlPriceType in the Security Definition (tag 35-MsgType=d) message indicates the settlement price type.

On Sunday start, the Security Definition (tag 35-MsgType=d) message is published initially to the Instrument Definition feeds and the Market Data Incremental Refresh feeds without tag 5796-TradingReferenceDate. The settlement price in tag 1150-TradingReferencePrice is published on Sunday start, but will reflect the last published settlement price from the previous trading week.

CME Globex sends three types of settlement prices in the Market Data Incremental Refresh (tag 35-MsgType=X) message:

- Actual Preliminary Settlement
- Net Change Preliminary Settlement
- Final Settlement.

The e-mini, e-micro, and miNY products use the preliminary settlement price of their corresponding full-sized product, When the full-sized product has a smaller minimum tick increment than the corresponding e-mini, e-micro, or miNY product, the preliminary settlement price may be off-tick. To accommodate this off-tick preliminary settlement price, all products that settle to another product's settlement price (the e-mini, miNY, and micro products) are evaluated for settlement price rounding and the resulting Net Change Preliminary settlement price, rounded if required, is then sent to customers. For products that do not have a settlement price rounding evaluation, tag 286-OpenCloseSettleFlag=101 indicates an Actual Preliminary settlement price and tag 286-OpenCloseSettleFlag=100 indicates Net Change Preliminary settlement price.

In the following example, the E-mini S&P 500 Future settles to the S&P 500 Future settlement price:

Product	Minimum Tick Increment
S&P 500 Future (tag 1151-SecurityGroup=SP)	0.10
E-mini S&P 500 Future (tag 1151-SecurityGroup=ES)	0.25

1. The S&P 500 Future settles at 1213.60.
2. The E-mini S&P 500 Future settlement price of 1213.60 is sent and marked as an Actual preliminary settlement price (tag 286-OpenCloseSettleFlag=101).
3. The settlement price 1213.60 is evaluated for rounding and must be rounded to 1213.50 to align with the E-mini S&P 500 Future minimum tick increment.
4. The E-mini S&P 500 Future settlement price of 1213.50 is sent and marked as a Net Change preliminary settlement price (tag 286-OpenCloseSettleFlag=100).

When the Actual preliminary settlement price is on-tick for the e-mini, e-micro, and miNY products, CME Globex will continue to send a Net Change preliminary settlement price with the identical settlement price. For example:

1. The S&P 500 Future settles at 1213.50.
2. The E-mini S&P 500 Future settlement price of 1213.50 is sent and marked as an Actual preliminary settlement price (tag 286-OpenCloseSettleFlag=101).
3. The settlement price 1213.50 is evaluated for rounding and is not required to align with the E-mini S&P 500 Future minimum tick increment.
4. The E-mini S&P 500 Future settlement price of 1213.50 is sent and marked as a Net Change preliminary settlement price (tag 286-OpenCloseSettleFlag=100).

All other ("non-rounded") products are settled to their own trade prices and are always on-tick. For these products, only the Actual Preliminary is sent. For non-rounded products, that is, those that do not have a settlement price rounding evaluation, tag 286-OpenCloseSettleFlag=100 indicates an Actual Preliminary settlement price. Tag 286-OpenCloseSettleFlag=101 is not sent for these products since their settlement prices are always on-tick.

Refer to the following link for more information on settlement prices: http://www.cmegroup.com/market-data/settlements/.

The following tables show the tags that are included in the Market Data Incremental Refresh (tag 35-MsgType=X) message for each settlement price type.

Table 6.4. E-mini, e-micro, and miNY Products: Net Change Preliminary Price

Tag	FIX Name	Format	Sample Value	Description
75	TradeDate	LocalMkt Date (8)	20100615	Indicates current session date in YYYYMMDD format.
64	SettleDate	LocalMkt Date (8)	20100614	Settlement Date to which this price applies in YYYYMMDD format.
286	OpenCloseSettle Flag	Int (3)	100	Flag to indicate the type of entry 100 = Net Change Preliminary
269	MDEntryType	Char (1)	6	Type of Market Data entry. 6 = Settlement Price

Table 6.5. E-mini, e-micro, and miNY Products: Actual Preliminary Price

Tag	FIX Name	Format	Sample Value	Description
75	TradeDate	LocalMkt Date (8)	20100615	Indicates current session date in YYYYMMDD format.
64	SettleDate	LocalMkt Date (8)	20100614	Settlement Date to which this price applies in YYYYMMDD format.
286	OpenCloseSettle Flag	Int (3)	101	Flag to indicate the type of entry 101 = Actual Preliminary
269	MDEntryType	Char (1)	6	Type of Market Data entry. 6 = Settlement Price

Table 6.6. Non-rounded Products: Preliminary Settlement

Tag	FIX Name	Format	Sample Value	Description
75	TradeDate	LocalMkt Date (8)	20100615	Indicates current session date in YYYYMMDD format.
64	SettleDate	LocalMkt Date (8)	20100614	Settlement Date to which this price applies in YYYYMMDD format.

Table 6.6. Non-rounded Products: Preliminary Settlement

Tag	FIX Name	Format	Sample Value	Description
286	OpenCloseSetleFlag	Int (3)	100	Flag to indicate the type of entry 100 = Actual Preliminary
269	MDEntryType	Int (1)	6	Type of Market Data entry. 6 = Settlement Price

Table 6.7. Final Settlement

Tag	FIX Name	Format	Sample Value	Description
75	TradeDate	LocalMkt Date (8)	20100309	Indicates session date of trade referenced in this message in YYYYMMDD format.
64	SettleDate	LocalMkt Date (8)	20100309	Settlement Date for the instrument. The format is YYYYMDD
269	MDEntryType	Int (1)	6	Type of Market Data entry. 6 = Settlement Price

Note: **The final settlement will not contain tag 286-OpenCloseSettleFlag.**

For the settlement prices updates in the Security Definition (tag 35-MsgType=d) message, tag 731-SettlPriceType indicates the settlement price type.

Table 6.8. Security Definition (tag 35-MsgType=d) Message

Tag	FIX Name	Format	Valid Values	Description
731	SettlPriceType	String (3)	1=Final 100 = Actual Preliminary for instruments not subject to settlement price rounding; 100 = Net Change Preliminary for instruments subject to settlement price rounding (e-mini, e-micro, and miNY products) 101 = Actual Preliminary for instruments subject to settlement price rounding (e-mini, e-micro and miNY products)	Indicates whether settlement price is preliminary or final.

Note: **For instruments subject to settlement price rounding and having an on-tick Actual Preliminary, the settlement price is the same for tag 731-SettlPriceType=100 and tag 731-SettlePriceType=101.**

6.11 Trading Statistics Reset Based on Market State Value

Statistics are sent on the Incremental Feed with the Market Data Incremental Refresh (tag 35-MsgType=X) message as they occur throughout a trading session. Certain statistics are also made available on the Market Recovery Feed with the Market Data Snapshot Full Refresh (tag 35-MsgType=W) message. Statistics are removed at the end of a trading session from the Market Recovery Feed on the Market Data Snapshot Full Refresh (tag 35-MsgType=W) message.

Tag 1174-SecurityTradingEvent=4 in the Security Status (tag 35-MsgType=f) Message indicates the end of a trading session, and triggers the process to remove statistics from the Market Recovery Feed on the Market Data Snapshot Full Refresh (tag 35-MsgType=W) message.

The following process takes place:

1. A Security Status (tag 35-MsgType=f) message is sent, containing the following tags:
   - Tag 326-SecurityTradingStatus set to 18 = Not available for trading (end of session)
   - Tag 1174-SecurityTradingEvent set to 4 = End of trading
2. CME Globex removes the following statistics from the Market Recovery Feed on the Market Data Snapshot Full Refresh (tag 35-MsgType=W) message:
   - Session High Bid (tag 269-MDEntryType=N)

- Session Low Offer (tag 269-MDEntryType=O)
- Opening Price (tag 269-MDEntryType=4)
- Trading Session High Price (tag 269-MDEntryType=7)
- Trading Session Low Price (tag 269-MDEntryType=8)
- Trade Volume (tag 1020-TradeVolume) for the Last Trade data block (tag 269-MDEntryType=2)

Note: **The Last Trade (tag 269-MDEntryType=2) statistic is not removed from the Market Recovery Feed. It is not removed until a new Last Trade is generated.**

3. Client systems can choose to reset the statistics.

   Statistics will not be sent in the Market Data Snapshot Full Refresh (tag 35-MsgType=W) message on the Market Recovery feed until activity on the Incremental feed creates new statistics.

Note: **Tag 75-TradeDate indicates the session date of the trade referenced in this message.**

Note: **Tag 64-SettleDate indicates the settlement date for the price in tag 1150-TradingReference-Price.**

7. CME Globex Pricing

This section describes the pricing and tick convention used for instruments available on CME Globex. The CME Globex price is the format used in all price tags in order entry iLink and FIX/FAST market data MDP interfaces. One method for obtaining the CME Globex price is from tag 270-MDEntryPX of the Market Data Incremental Refresh tag 35-MsgType=X message.

7.1 Tick Size Calculation

The tick is the minimum price fluctuation allowed for a futures or options contract during a trading session as specified by the contract terms of CME Group. An instrument can have either a variable tick (primarily for options instruments) or a standard tick. Both the variable tick and standard tick are obtained from the Security Definition (tag 35-MsgType=d) message as described below.

7.1.1 Variable Tick Instrument

When tag 871-InstAttribType=23 in the Security Definition (tag 35-MsgType=d) message, the instrument is VTT (Variable Tick Table)-eligible. The following process should be performed to calculate the tick size for a VTT eligible instrument. Tag 872-InstAttribValue will contain the VTT index code as shown in the “Variable Tick Table” on page 88.

Note: **Tag 969-MinPriceIncrement will not be sent in the Security Definition (tag 35-MsgType=d) message for VTT-eligible instruments.**

7.1.1.1 Example - Variable Tick Instrument

CME Globex sends the following Security Definition (tag 35-MsgType=d) message for an instrument with a price of 510:

1128=9|35=d|49=CME|34=9647|52=20080608171603359|15=USD|22=8|48=300998|55=OS|107=SPM8P1750|200=200806|202=175000|207=XCME|461=OPAFPS|462=5|562=1|711=1|311=[N/A]|305=8|309=2536|827=0|864=2|865=5|866=20060707|1145=202500000|865=7|866=20080619|1145=202500000|870=5|**871=23|872=01|**871=24|872=1|871=24|872=2|871=24|872=3|871=24|872=14|947=USD|1140=9999|1141=1|1022=GBX|264=1|1143=600|1146=0|1147=0|1148=5|1149=9999900|1150=36530|1151=SP|1180=8|9787=0.01|9850=5

1. Tag 871-InstAttribType=23, so the instrument is VTT eligible instrument.
2. Tag 872-InstAttribValue contains the VTT index code 01, as shown in the Table 7.1 on page 88.
3. Since tag 872-InstAttribValue is 01 for a market price of 510 (P=510>500), the tick size for the contract is 10.

Table 7.1. Variable Tick Table

Variable Tick Table Index Code	Current CME Globex Price (P)	CME Globex Tick Size
01	P < -500	10
01	-500 ≤ P ≤ 500	5
01	P > 500	10

Table 7.1. Variable Tick Table

Variable Tick Table Index Code	Current CME Globex Price (P)	CME Globex Tick Size
02	-5 ≤ P ≤ 5	0.5
02	P > 5	1
03	-10 ≤ P ≤ 10	1
03	P > 10	2
04	P < -500	25
04	-500 ≤ P ≤ 500	5
04	P > 500	25
09	The 09 Variable Tick Index Code indicates that the spread is eligible to trade in 0.5 (half-tick) increments when any of the leg settlement prices are equal to or less than 5.0 full ticks.	
10	P < -300	25
10	-300 ≤ P ≤ 300	5
10	P > 300	25
11	P < -300	10
11	-300 ≤ P ≤ 300	5
11	P > 300	10
12	P < -5	0.50
12	-5 ≤ P ≤ 5	0.25
12	P > 5	0.50

7.1.2 Standard Tick Instrument

If the Security Definition (tag 35-MsgType=d) message for an instrument does not contain tag 871-InstAttribType=23, the instrument uses a standard tick. The value found in tag 969-MinPriceIncrement is the tick size.

7.1.2.1 Example - Standard Tick Instrument

CME Globex sends the following Security Definition (tag 35-MsgType=d) message for a standard tick instrument:

1128=9|9=425|35=d|49=CME| 34=13591|52=20080522221850254|15=USD|22=8|48=806800|55=GE|107=GEM8|200=200806|207=XCME|461=FFDXSX|462=14|562=1|827=2|864=2|865=5|866=19980617|1145=213000000|865=7|866=20080616|1145=100000000|870=5|871=24|872=1|871=24|872=2|871=24|872=3|871=24|872=4|871=24|872=5|947=USD|**969=0.5**|1140=30000|1141=1|1022=GBX|264=5|1142=A|

1143=10.00|1144=3|1146=12.5000|1148=9559.50|1149=9959.50|1150=9759.50|1151=GE|1180=9|9787=0.01|10=216

The tick size is 0.5.

7.2 Suggested Display Format for Non-Fractional Products

Note: **The resulting display of price and tick below are suggested by CME Group but are not required.**

The client system can use the tags shown in the following table to display the price of the instrument. The display factor is multiplied by both the CME Globex Tick and the CME Globex Price conventions to calculate the display tick and display price. The following table is a sample display of CME Globex ticks and prices.

Contract	CME Globex Tick (tag 969-MinPriceIncrement)	CME Globex Price (tag 270-MDEntryPx)	Display Factor (tag 9787-DisplayFactor)	Display Tick	Display Price
ESH6	25	113700	0.01	.25	1137.00
GEM6	.5	9886.5	0.01	.005	98.865

CME Globex Price x Display Factor = Display Price

CME Globex Tick x Display Factor = Display Tick

7.3 Fractional Pricing

The information in this section is for products that require a decimal-to-fractional price conversion. These products are identified in the Security Definition (tag 35-MsgType=d) message by the following tags:

- tag 871-InstAttributeType=’24’
- tag 872-InstAttribValue=’12’
- Products that do not require a decimal-to-fractional price conversion will not contain tag 871-InstAttributeType=’24’ and tag 872-InstAttribValue=’12’. For those products, customers can use the Display Factor (tag 9787-DisplayFactor) for FIX/FAST to determine delimiter placement.

Note: **Display Factor (tag 9787-DisplayFactor) should not be used for fractionals.**

7.3.1 MDP FIX/FAST Format

Fractional price data in FIX/FAST is transmitted in four repeating groups of tag 871-InstAttributeType and tag 872-InstAttribValue. The first repeating group indicates that a fractional display price is present, the second repeating group indicates main fraction denominator, the third repeating group indicates the sub fraction denominator, and the fourth repeating group indicates the number of decimals in the displayed price. These repeating groups function as follows:

Tag	FIX Name	Description
➔871	InstAttributeType	'24' = Eligibility
➔872	InstrAttribValue	'12' = When the preceding tag 871-InstAttributeType = '24' and this tag = '12' this indicates the product has a fractional price.
➔871	InstAttributeType	'25' = Main Fraction
➔872	InstrAttribValue	When the preceding tag 871-InstAttributeType = '25' this tag will contain the main fraction denominator.
➔871	InstAttributeType	'26' = Sub Fraction
➔872	InstrAttribValue	When the preceding tag 871-InstAttributeType = '26' this tag will contain the sub fraction denominator.
➔871	InstAttributeType	'27' = Display Price Precision
➔872	InstrAttribValue	When the preceding tag 871-InstAttributeType = '27' this tag will contain the number of decimals in the displayed price.

FIX/FAST Tick Display Format Repeating Group	
Tag 871-InstAttributeType=26	Tag 871-InstAttributeType=25
Tag 872-InstAttributeValue=numerator	Tag 872-InstAttributeValue=denominator
tag 872-InstAttributeValue is not sent for these values	1
	2
	4
	8
	16
	32
	64
2	32
4	32
2	64

7.3.1.1 Example - ZNZ9 - 10 Year U.S. Treasury Note Future

1128=9|9=455|35=d|49=CME|34=523|52=20080525164027159|15=USD|22=8|48=500268|55=ZN|107=ZNZ9|200=200812|207=XCBT|461=FFDXSX|462=14|562=1|827=2|864=2|865=5|866=20070919|1145=213000000|865=7|866=20081219|1145=180100000|870=9|871=24|872=1|871=24|872=3|871=24|872=4|871=24|872=6|871=24|872=7|**871=24|872=12|871=25|872=32|871=26|872=2|871=27|872=3|**947=USD|969=0.015625|1140=49999|1141=110|22=GBX|264=5|1142=f|1143=0.937500|1146=0.001000|1150=113.718750|1151=ZN|1180=115|9787=1

TDFT= Tick Display Format Type (Numerator and Denominator)

NDDP = Number of Decimals in the Display Price

10 Yr U.S. Treasury Note Future, TDFT Half 1/32 Tick, NDDP 3
Original decimal price 112.625

Fractional part after conversion 20 (.625 x 32)

Integer part and fractional part 112 20

Apply NDDP of 03

112 20

123 (Display places)

Resulting formatted price 112’200

7.3.2 Decimal to Formatted Fractional Tick

FPF	1	1	1	1	1	1	1	1	1	1
TDFT	01	02	04	08	16	32	64	Half 1/32 Tick	Quarter 1/32 Tick	Half 1/64 Tick
NDDP	00	01*	01	01	02	02	02	03	03	03
0			0	0	00	00	00	000	000	000
.0078125									002	005
.015625							01	005	005	010
.0234375									007	015
.03125						01	02	010	010	020
.0390625									012	025
.046875							03	015	015	030
.0546875									017	035
.0625					01	02	04	020	020	040
.0703125									022	045
.078125							05	025	025	050
.0859375									027	055
.09375						03	06	030	030	060
.1015625									032	065
.109375							07	035	035	070
.1171875									037	075
.125				1	02	04	08	040	040	080
.1328125									042	085
.140625							09	045	045	090
.1484375									047	095
.15625						05	10	050	050	100
.1640625									052	105
.171875							11	055	055	110
.1796875									057	115
.1875					03	06	12	060	060	120
.1953125									062	125
.203125							13	065	065	130
.2109375									067	135
.21875						07	14	070	070	140
.2265625									072	145
.234375							15	075	075	150
.2421875									077	155
.25			2	2	04	08	16	080	080	160
.2578125									082	165
.265625							17	085	085	170
.2734375									087	175
.28125						09	18	090	090	180
TDFT	01	02	04	08	16	32	64	Half 1/32 Tick	Quarter 1/32 Tick	Half 1/64 Tick
NDDP	00	01*	01	01	02	02	02	03	03	03
.2890625									092	185

.296875							19	095	095	190
.3046875									097	195
.3125					05	10	20	100	100	200
.3203125									102	205
.328125							21	105	105	210
.3359375									107	215
.34375						11	22	110	110	220
.3515625									112	225
.359375							23	115	115	230
.3671875									117	235
.375				3	06	12	24	120	120	240
.3828125									122	245
.390625							25	125	125	250
.3984375									127	255
.40625						13	26	130	130	260
.4140625									132	265
.421875							27	135	135	270
.4296875									137	275
.4375					07	14	28	140	140	280
.4453125									142	285
.453125							29	145	145	290
.4609375									147	295
.46875						15	30	150	150	300
.4765625									152	305
.484375							31	155	155	310
.4921875									157	315
5			5	4	08	16	32	160	160	320
.5078125									162	325
.515625							33	165	165	330
.5234375									167	335
.53125						17	34	170	170	340
.5390625									172	345
.546875							35	175	175	350
.5546875									177	355
.5625					09	18	36	180	180	360
.5703125									182	365
.578125							37	185	185	370
.5859375									187	375
.59375						19	38	190	190	380
.6015625									192	385
.609375							39	195	195	390
.6171875									197	395
.625				5	10	20	40	200	200	400
.6328125									202	405
.640625							41	205	205	410
TDFT	01	02	04	08	16	32	64	Half 1/32 Tick	Quarter 1/32 Tick	Half 1/64 Tick
NDDP	00	01*	01	01	02	02	02	03	03	03

.6484375									207	415
.65625						21	42	210	210	420
.6640625									212	425
.671875							43	215	215	430
.6796875									217	435
.6875					11	22	44	220	220	440
.6953125									222	445
.703125							45	225	225	450
.7109375									227	455
.71875						23	46	230	230	460
.7265625									232	465
.734375							47	235	235	470
.7421875									237	475
.75			7	6	12	24	48	240	240	480
.7578125									242	485
.765625							49	245	245	490
.7734375									247	495
.78125						25	50	250	250	500
.7890625									252	505
.796875							51	255	255	510
.8046875									257	515
.8125					13	26	52	260	260	520
.8203125									262	525
.828125							53	265	265	530
.8359375									267	535
.84375						27	54	270	270	540
.8515625									272	545
.859375							55	275	275	550
.8671875									277	555
.875				7	14	28	56	280	280	560
.8828125									282	565
.890625							57	285	285	570
.8984375									287	575
.90625						29	58	290	290	580
.9140625									292	585
.921875							59	295	295	590
.9296875									297	595
.9375					15	30	60	300	300	600
.9453125									302	605
.953125							61	305	305	610
.9609375									307	615
.96875						31	62	310	310	620
.9765625									312	625
.984375							63	315	315	630
.9921875									317	635
1			0	0	00	00	00	000	000	000

7.4 Formatted Fractional Tick to Decimal

TDFT	01
NDDP	00
Ticks	Decimal

TDFT	02
NDDP	01*
Ticks	Decimal

TDFT	04
NDDP	01
Ticks	Decimal
0	.00
2	.25
5	.50
7	.75

TDFT	08
NDDP	01
Ticks	Decimal
0	.000
1	.125
2	.250
3	.375
4	.500
5	.625
6	.750
7	.875

TDFT	16
NDDP	02
Ticks	Decimal
00	.0000
01	.0625
02	.1250
03	.1875
04	.2500
05	.3125
06	.3750
07	.4375
08	.5000
09	.5625
10	.6250
11	.6875
12	.7500
13	.8125
14	.8750
15	.9375

TDFT	32
NDDP	02
Ticks	Decimal
00	.00000
01	.03125
02	.06250
03	.09375
04	.12500
05	.15625
06	.18750
07	.21875
08	.25000
09	.28125
10	.31250
11	.34375
12	.37500
13	.40625
14	.43750
15	.46875

TDFT	32
NDDP	02
Ticks	Decimal
16	.50000
17	.53125
18	.56250
19	.59375
20	.62500
21	.65625
22	.68750
23	.71875
24	.75000
25	.78125
26	.81250
27	.84375
28	.87500
29	.90625
30	.93750
31	.96875

FPF	1
TDFT	64
NDDP	02
Ticks	Decimal
00	.000000
01	.015625
02	.031250
03	.046875
04	.062500
05	.078125
06	.093750
07	.109375
08	.125000
09	.140625
10	.156250
11	.171875
12	.187500
13	.203125
14	.218750
15	.234375
16	.250000
17	.265625
18	.281250
19	.296875
20	.312500
21	.328125
22	.343750
23	.359375
24	.375000
25	.390625
26	.406250
27	.421875
28	.437500
29	.453125
30	.468750
31	.484375

FPF	1
TDFT	64
NDDP	02
Ticks	Decimal
32	.500000
33	.515625
34	.531250
35	.546875
36	.562500
37	.578125
38	.593750
39	.609375
40	.625000
41	.640625
42	.656250
43	.671875
44	.687500
45	.703125
46	.718750
47	.734375
48	.750000
49	.765625
50	.781250
51	.796875
52	.812500
53	.828125
54	.843750
55	.859375
56	.875000
57	.890625
58	.906250
59	.921875
60	.937500
61	.953125
62	.968750
63	.984375

FPF	1
TDFT	Half 1/32 Tick
NDDP	03
Ticks	Decimal
000	.000000
005	.015625
010	.031250
015	.046875
020	.062500
025	.078125
030	.093750
035	.109375
040	.125000
045	.140625
050	.156250
055	.171875
060	.187500
065	.203125
070	.218750
075	.234375
080	.250000
085	.265625
090	.281250
095	.296875
100	.312500
105	.328125
110	.343750
115	.359375
120	.375000
125	.390625
130	.406250
135	.421875
140	.437500
145	.453125
150	.468750
155	.484375

FPF	1
TDFT	Half 1/32 Tick
NDDP	03
Ticks	Decimal
160	.500000
165	.515625
170	.531250
175	.546875
180	.562500
185	.578125
190	.593750
195	.609375
200	.625000
205	.640625
210	.656250
215	.671875
220	.687500
225	.703125
230	.718750
235	.734375
240	.750000
245	.765625
250	.781250
255	.796875
260	.812500
265	.828125
270	.843750
275	.859375
280	.875000
285	.890625
290	.906250
295	.921875
300	.937500
305	.953125
310	.968750
315	.984375

FPF	1
TDFT	Quarter 1/32 Tick
NDDP	03
Ticks	Decimal
000	.0000000
002	.0078125
005	.0156250
007	.0234375
010	.0312500
012	.0390625
015	.0468750
017	.0546875
020	.0625000
022	.0703125
025	.0781250
027	.0859375
030	.0937500
032	.1015625
035	.1093750
037	.1171875
040	.1250000
042	.1328125
045	.1406250
047	.1484375
050	.1562500
052	.1640625
055	.1718750
057	.1796875
060	.1875000
062	.1953125
065	.2031250
067	.2109375
070	.2187500
072	.2265625
075	.2343750
077	.2421875

FPF	1
TDFT	Quarter 1/32 Tick
NDDP	03
Ticks	Decimal
080	.2500000
082	.2578125
085	.2656250
087	.2734375
090	.2812500
092	.2890625
095	.2968750
097	.3046875
100	.3125000
102	.3203125
105	.3281250
107	.3359375
110	.3437500
112	.3515625
115	.3593750
117	.3671875
120	.3750000
122	.3828125
125	.3906250
127	.3984375
130	.4062500
132	.4140625
135	.4218750
137	.4296875
140	.4375000
142	.4453125
145	.4531250
147	.4609375
150	.4687500
152	.4765625
155	.4843750
157	.4921875

FPF	1
TDFT	Quarter 1/32 Tick
NDDP	03
Ticks	Decimal
160	.5000000
162	.5078125
165	.5156250
167	.5234375
170	.5312500
172	.5390625
175	.5468750
177	.5546875
180	.5625000
182	.5703125
185	.5781250
187	.5859375
190	.5937500
192	.6015625
195	.6093750
197	.6171875
200	.6250000
202	.6328125
205	.6406250
207	.6484375
210	.6562500
212	.6640625
215	.6718750
217	.6796875
220	.6875000
222	.6953125
225	.7031250
227	.7109375
230	.7187500
232	.7265625
235	.7343750
237	.7421875

FPF	1
TDFT	Quarter 1/32 Tick
NDDP	03
Ticks	Decimal
240	.7500000
242	.7578125
245	.7656250
247	.7734375
250	.7812500
252	.7890625
255	.7968750
257	.8046875
260	.8125000
262	.8203125
265	.8281250
267	.8359375
270	.8437500
272	.8515625
275	.8593750
277	.8671875
280	.8750000
282	.8828125
285	.8906250
287	.8984375
290	.9062500
292	.9140625
295	.9218750
297	.9296875
300	.9375000
302	.9453125
305	.9531250
307	.9609375
310	.9687500
312	.9765625
315	.9843750
317	.9921875

Formatted Fractional Tick to Decimal

FPF	1	FPF	1	FPF	1	FPF	1
TDFT	Half 1/64 Tick	TDFT	Half 1/64 Tick	TDFT	Half 1/64 Tick	TDFT	Half 1/64 Tick
NDDP	03	NDDP	03	NDDP	03	NDDP	03
Ticks	Decimal	Ticks	Decimal	Ticks	Decimal	Ticks	Decimal
000	.0000000	160	.2500000	320	.5000000	480	.7500000
005	.0078125	165	.2578125	325	.5078125	485	.7578125
010	.0156250	170	.2656250	330	.5156250	490	.7656250
015	.0234375	175	.2734375	335	.5234375	495	.7734375
020	.0312500	180	.2812500	340	.5312500	500	.7812500
025	.0390625	185	.2890625	345	.5390625	505	.7890625
030	.0468750	190	.2968750	350	.5468750	510	.7968750
035	.0546875	195	.3046875	355	.5546875	515	.8046875
040	.0625000	200	.3125000	360	.5625000	520	.8125000
045	.0703125	205	.3203125	365	.5703125	525	.8203125
050	.0781250	210	.3281250	370	.5781250	530	.8281250
055	.0859375	215	.3359375	375	.5859375	535	.8359375
060	.0937500	220	.3437500	380	.5937500	540	.8437500
065	.1015625	225	.3515625	385	.6015625	545	.8515625
070	.1093750	230	.3593750	390	.6093750	550	.8593750
075	.1171875	235	.3671875	395	.6171875	555	.8671875
080	.1250000	240	.3750000	400	.6250000	560	.8750000
085	.1328125	245	.3828125	405	.6328125	565	.8828125
090	.1406250	250	.3906250	410	.6406250	570	.8906250
095	.1484375	255	.3984375	415	.6484375	575	.8984375
100	.1562500	260	.4062500	420	.6562500	580	.9062500
105	.1640625	265	.4140625	425	.6640625	585	.9140625
110	.1718750	270	.4218750	430	.6718750	590	.9218750
115	.1796875	275	.4296875	435	.6796875	595	.9296875
120	.1875000	280	.4375000	440	.6875000	600	.9375000
125	.1953125	285	.4453125	445	.6953125	605	.9453125
130	.2031250	290	.4531250	450	.7031250	610	.9531250
135	.2109375	295	.4609375	455	.7109375	615	.9609375
140	.2187500	300	.4687500	460	.7187500	620	.9687500
145	.2265625	305	.4765625	465	.7265625	625	.9765625
150	.2343750	310	.4843750	470	.7343750	630	.9843750
155	.2421875	315	.4921875	475	.7421875	635	.9921875

8. Recovery

CME Group offers several options for recovering missed messages or synchronizing client systems to the latest state: Market Recovery, Instrument Replay, and TCP Replay. Instrument level sequencing and natural refresh can be utilized to supplement the recovery process.

This section contains an overview of the various recovery methods and supplemental methods.

- “Recovery Feeds” on Page 100 - this section describes the following: Market Recovery, Instrument Replay, and TCP Replay.
- “Using the Incremental Market Data Feed to Determine State” on Page 108 - the methods in this section provide supplemental methods to the recovery process, however they are not guaranteed to completely recover missed messages or synchronize client systems to the latest state.
- “Order Book Reset” on Page 112 - this section describes the process for synchronizing order books in the unlikely event of a CME Group dual component failure. In this scenario, order books on the channel may be “corrupted”.

There is also a section that describes the process to recover in various failure scenarios.

- “Recovering Data - Process” on Page 102

Note: **CME Group strongly recommends that client systems process both the A and B Incremental UDP feeds. UDP Feed A and UDP Feed B provide the first level of protection against missed market data messages.**

Note: **CME Group recommends Market Recovery in conjunction with Natural Refresh as a primary recovery option.**

Note: **All customers must certify for Market Recovery functionality.**

8.1 Recovery Feeds

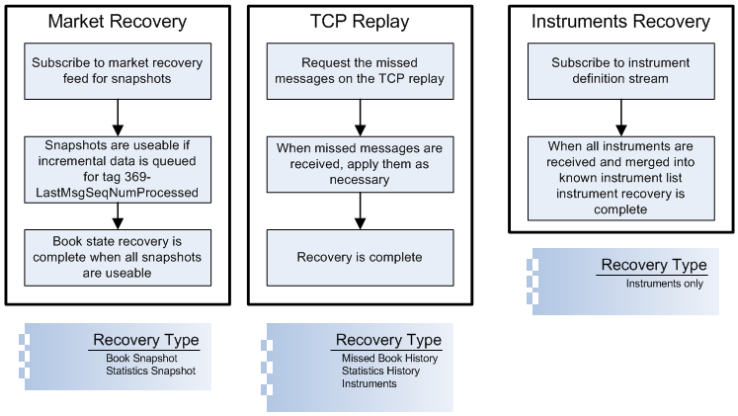
UDP Feed A and UDP Feed B are used to disseminate CME Group incremental market data using bandwidth efficient FAST encrypted FIX messages. All FIX message types are sent through UDP Feed A and UDP Feed B applicable Market Data Groups (book update, statistics, quotes, instrument definitions, trades, instrument / instrument group, CME Globex status). Please note that a single FIX/FAST message can contain multiple updates for multiple instruments. In some instances, a market data message may be missed from UDP Feed A but will be received from UDP Feed B. For more information, refer to “Incremental UDP Feed A and B” on Page 7.

In some instances, messages may be missed on both feeds, requiring a recovery process to take place. Message loss is detected using the decoded FIX message sequence numbers (tag 34-MsgSeqNum), which are also found in the Preamble sequence number. The message sequence number is an incrementing number, therefore, if a gap is detected between messages in the tag 34-MsgSeqNum value, or the Preamble sequence number, this indicates a message has been missed. In addition, tag 83-RptSeq can be used to detect a gap between the messages at the instrument level. It should be assumed at this point that all books maintained in the client system may no longer have the correct, latest state maintained by CME Group. Client systems must resynchronize all books to the latest state maintained by CME Group, and determine whether any new instrument definitions were published. During this synchronization process, all books are initially assumed to be in an incorrect state and are recovered during the synchronization process.

All book updates in the latest Market Data Incremental Refresh message (tag 35-MsgType=X) messages must be processed before the order book can be considered valid.

The following options are offered by CME Group to recover missed messages or synchronize client systems to the latest state:

- “Market Recovery Overview” on Page 100
- “Instrument Replay Overview” on Page 104
- “TCP Replay Overview” on Page 104



8.1.1 Market Recovery Overview

This recovery method should be used for large-scale data recovery (i.e. major outage or late joiners) to synchronize client systems to the latest state maintained by CME Group. Client systems can use the Market Recovery feed on each channel to determine the state of each book in every channel. Each Market Recovery feed constantly loops and sends the Market Data Snapshot Full Refresh (tag 35-MsgType = W) message.

The Market Recovery feed is known to be valid as of an instrument level sequence number on the Incremental Market Data feed. This instrument level sequence number (tag 83-RptSeq) is found on each Market Data Incremental Refresh Message (tag 35-MsgType=X) on the Incremental Market Data feed. It is also found on every Market Data Snapshot Full Refresh (tag 35-MsgType=W) message on the Market Recovery feed. As a result, it can be determined which book updates were missed for a specific instrument.

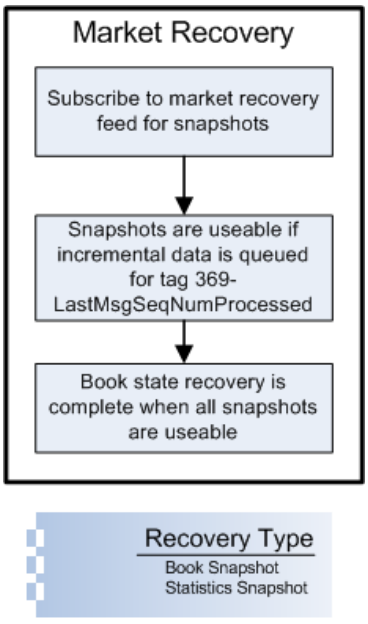
The Market Recovery feed is also known to be valid as of a sequence number on the Incremental Market Data feed. The sequence number (tag 369-LastMsgSeqNumProcessed) on the Market Recovery feed corresponds to the sequence number [(tag 34-MsgSeqNum) or the Preamble sequence number] on the Incremental Market Data feed. Tag 369-LastMsgSeqNumProcessed is contained in each Market Data Snapshot Full Refresh (tag 35=W) message and tag 34-MsgSeqNum is contained in each Market Data Incremental Refresh message. The sequence number can be used to determine which book updates were missed for a specific instrument.

If the Market Recovery method is used, client systems also need to subscribe to the Instrument Definition feed to determine whether any new instruments were defined. Client systems will recover the most recent statistics on the Market Recovery feed. Any intermediary statistics will not be recovered.

Market Recovery in conjunction with Natural Refresh provides exceptional performance and reliable recovery of market data.

Note: **Client systems should queue real-time data until all snapshot data is retrieved. The queued data should then be applied as necessary.**

Note: **CME Group requires that the Market Recovery feeds be used for recovery purposes only. Once client systems have retrieved recovery data, client systems should stop listening to the Market Recovery feeds.**

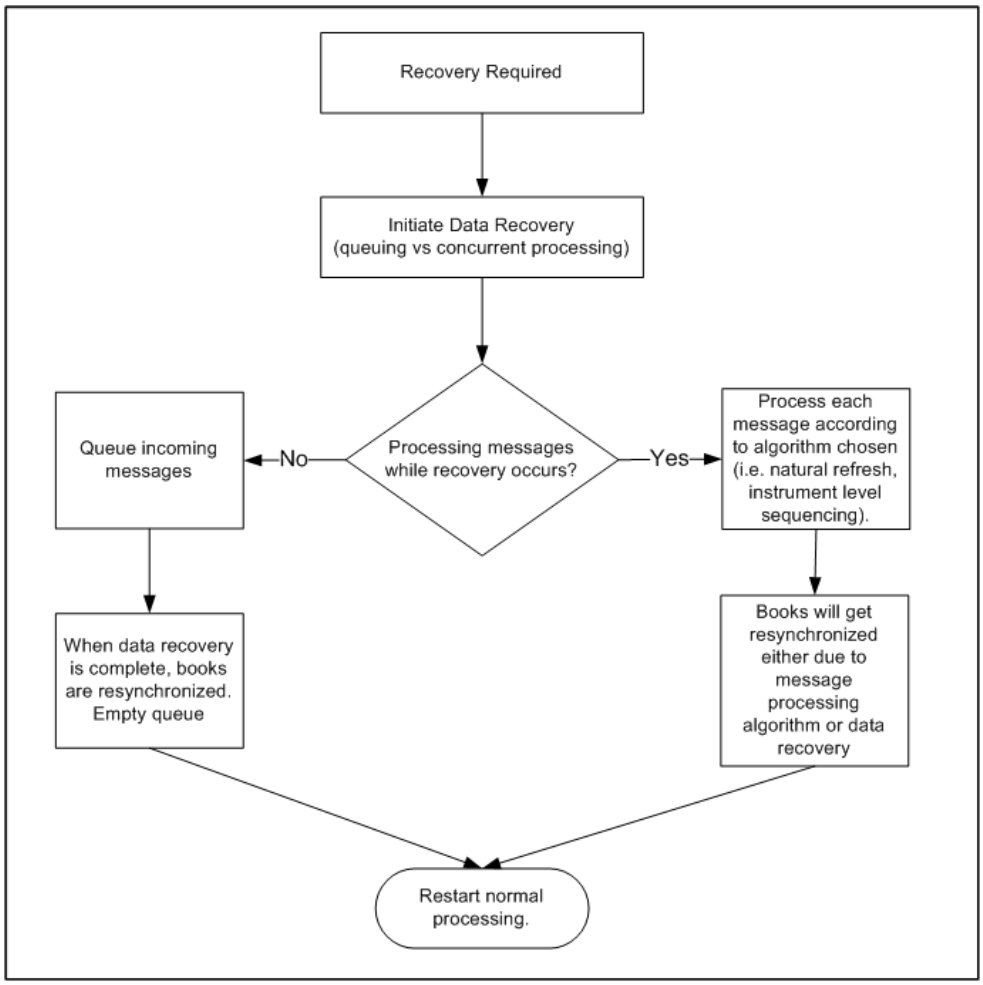


Note: **Tag 83-RptSeq provides an instrument level sequence number and can be leveraged instead of tag 369-LastMsgSeqNumProcessed.**

8.1.2 Recovering Data - Process

This section describes CME Group recommended recovery processes. Based on the recovery and supplemental methods, client systems can choose to implement those methods as applicable. In general, two paths can be followed: processing while recovering and queuing while recovering.

Note: **The recovering data process applies to affected channels only. Unaffected channels can continue normal processing.**



The following sections describe a process recovery options in various failure scenarios:

- “Large Scale Outage Using Market Recovery - Queuing” on Page 103
- “Large Scale Outage Using Market Recovery - Concurrent Processing” on Page 103

Note: **The Market Recovery feed should be used for large-scale data recovery (i.e. major outage or late joiners). TCP replay is not a performance-based recovery option and should only be used for small-scale data recovery.**

8.1.2.1 Large Scale Outage Using Market Recovery - Queuing

This section describes the process to follow for a large scale outage in which the client system is out of sync. This process uses Market Recovery - queuing of the Incremental Market Data feed and Market Recovery feed until the client system is synchronized to the latest state advertised by CME Group. In order to avoid an excessive number of queued messages, process snapshots and apply the applicable incremental feed as the snapshots arrive.

Note: **If the Market Recovery method is used, client systems also need to subscribe to the Instrument Definition feed to determine whether any new instruments were defined. Client systems will not recover any missed statistics on the Market Recovery feed.**

1. Identify channel(s) in which the client system is out of sync.
2. Listen to and queue the Incremental Market Data feed for the affected channel(s).
3. Listen to the Market Recovery feed for the affected channel(s).
4. Verify that all snapshots have been received for a given Market Recovery feed.

   Tag 911-TotNumReports in a Market Data Snapshot Full Refresh (tag 35-MsgType=W) message contains the total number of messages that have been sent. Listen to the Market Recovery feed until the indicated number of Market Data Snapshot Full Refresh (tag 35-MsgType=W) messages has been received.
5. Apply **all** incremental data in sequence, **where tag 34-MsgSeqNum (or the Preamble sequence number) is greater than the lowest value for tag 369-LastMsgSeqNumProcessed**

   OR

   where tag 83-RptSeq from the Market Data Incremental Refresh (tag 35-MsgType=X) message is greater than the lowest value for tag 83-RptSeq on the Market Recovery feed.
6. Using queued data, restart normal processing.

8.1.2.2 Large Scale Outage Using Market Recovery - Concurrent Processing

This section describes at the process to follow for a large scale outage using Market Recovery while continuing to process the Incremental Market Data feed and obtaining snapshots from the Market Recovery feed. Once books are recovered, they can resume normal processing even if other books are still being recovered.

Note: **If the Market Recovery method is used, client systems also need to subscribe to the Instrument Definition feed to determine whether any new instruments were defined.**

Client systems will not recover any missed statistics on the Market Recovery feed.

1. Identify channel(s) in which the client system is out of sync.
2. Listen to the Incremental Market Data feed for the affected channel(s) and optionally attempt a natural refresh.
3. Listen to the Market Recovery feed for the affected channel(s).

4. For each book, compare tag 369-LastMsgSeqNumProcessed on the Market Recovery feed to tag 34-MsgSeqNum (or the Preamble sequence number) on the Incremental Market Data feed and verify that the value for tag 34-MsgSeqNum is not lower

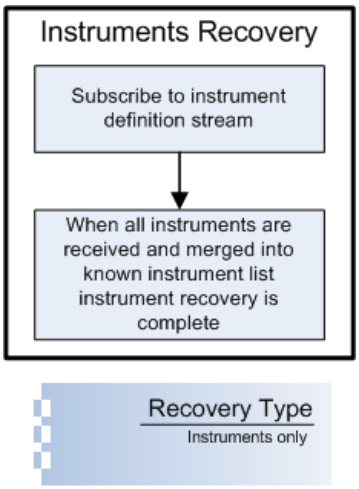
   OR

   compare tag 83-RptSeq on the Market Recovery feed to tag 83-RptSeq on the Incremental Market Data feed and verify that the value for tag 83-RptSeq on the Incremental Market Data feed is not lower.

5. Restart normal processing.

8.1.3 Instrument Replay Overview

Client systems can use the Instrument Replay feed on each channel to determine if any new instruments were defined through the Security Definition (tag 35-MsgType=d) message. The Instrument Replay feed constantly replays the current week's Security Definition (tag-MsgType=d) messages, and will reflect any additions, modifications, or deletes to CME Group contracts. Client systems should wait to begin the recovery process until tag 34-MsgSeqNum is 1 and tag 911-TotNumReports are at the last indicated number of reports is returned. CME Group does not guarantee the order of the messages being sent.



8.1.4 TCP Replay Overview

TCP replay should be used for small-scale data recovery (refer to "Implementation Considerations" on Page 106 for additional information). Client systems can recover specific messages that were missed using the sequence number and the TCP historical replay component. This method recovers all missed messages. **CME Group logs IP addresses, ports, and passwords of the originator and the Channel ID and range of the requested sequence number.**

Note: **TCP Replay is not a performance-based recovery option and should only be used if other options are unavailable.**

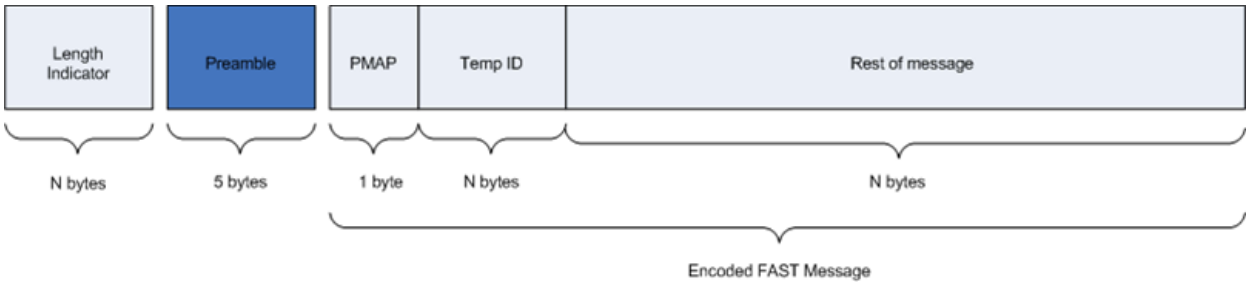
The TCP historical replay component allows you to request a replay of a set of messages already published on the UDP Incremental Market Data Channel. The request specifies messages to be replayed. The request is submitted with the FIX Market Data Request message (35=V). In response, CME Group sends the requested messages and a Logout (35=5) message. A Logout (35=5) message is also sent in case of rejection of the initial request, tag 58-Text will list the reason for the logout.

This type of request is sent through a new TCP connection established by the customer. When making the request, the channel and sequence number are specified. The responses are sent by CME Group through this same connection and the connection is then closed by CME Group once the replay is complete. A maximum of 2,000 messages can be requested at once.

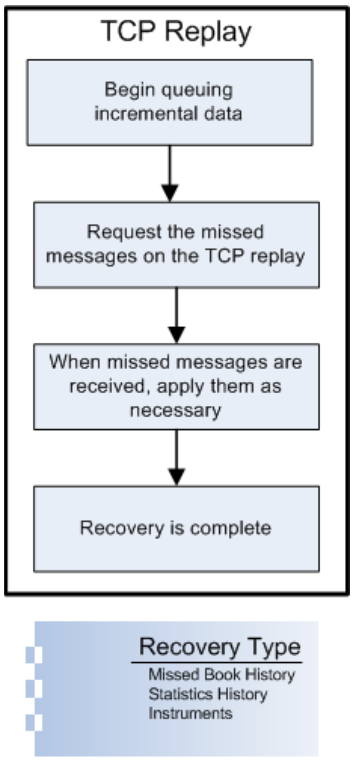
The TCP Replay feed contains 3 different message formats:

- **Raw FIX message** - Logon (tag 35-MsgType=A) - Customer to CME Group, Logout (tag 35-MsgType=5) - Customer to CME Group, and Market Data Request (tag 35-MsgType=V).
- **Length Indicator + Generated Preamble + FAST message** - Logon (tag 35-MsgType=A) - CME Group to Customer, Logout (tag 35-MsgType=5) - CME Group to Customer, Heartbeat (tag 35-MsgType=0).
- **Length Indicator + Original Preamble + FAST message -** All requested messages for recovery.

Note: **Replayed FIX/FAST messages will contain their original Preambles.**



Note: **Client systems should queue real-time data until all missed data is recovered. The recovered data should then be applied prior to queued data.**



8.1.4.1 Implementation Considerations

The following are limitations to consider prior to implementing TCP replay functionality:

- TCP Replay is not a performance-based recovery option and should only be used if other options are unavailable.
- TCP requests and responses are via TCP.
- Resend request messages and other customer-generated messages (i.e. heartbeat) are in FIX format, CME Globex-generated response messages are FAST encoded.
- There is a 24 hour time limit on the messages available via the TCP replay functionality.

Note: **Only one Market Data Request can be processed during the current established session. If multiple Market Data Requests are sent when the session is established, *only the first Market Data Request is processed and subsequent Market Data Requests are ignored*.**

To submit multiple market data requests, for each request, the client system must log in, submit the request, receive the messages, log out, and repeat until all requests are processed.

- The maximum number of messages that can be requested in one resend request message is 2000.

8.1.4.2 Process

1. Customer establishes a TCP connection.

- Refer to the configuration file for TCP IP and port information. For additional information on the configuration file, refer to “Services - Template Dissemination, Market Data Configuration, and Security Definition Flat File” on Page 9.

2. Customer sends Logon (tag 35-MsgType=A) message to CME Group.

Note: **Customer Username and Password are verified. If the Username and Password are incorrect, a Logout (tag 35-MsgType=5) will be sent.**

3. CME Group sends Logon (tag 35-MsgType=A) message to the customer.

Note: **CME Group will send a Logout (tag35-MsgType=5) message if a Market Data Request (tag35-MsgType=V) message is not received in 5 seconds.**

4. Customer sends Market Data Request (tag 35-MsgType=V) message to CME Group.

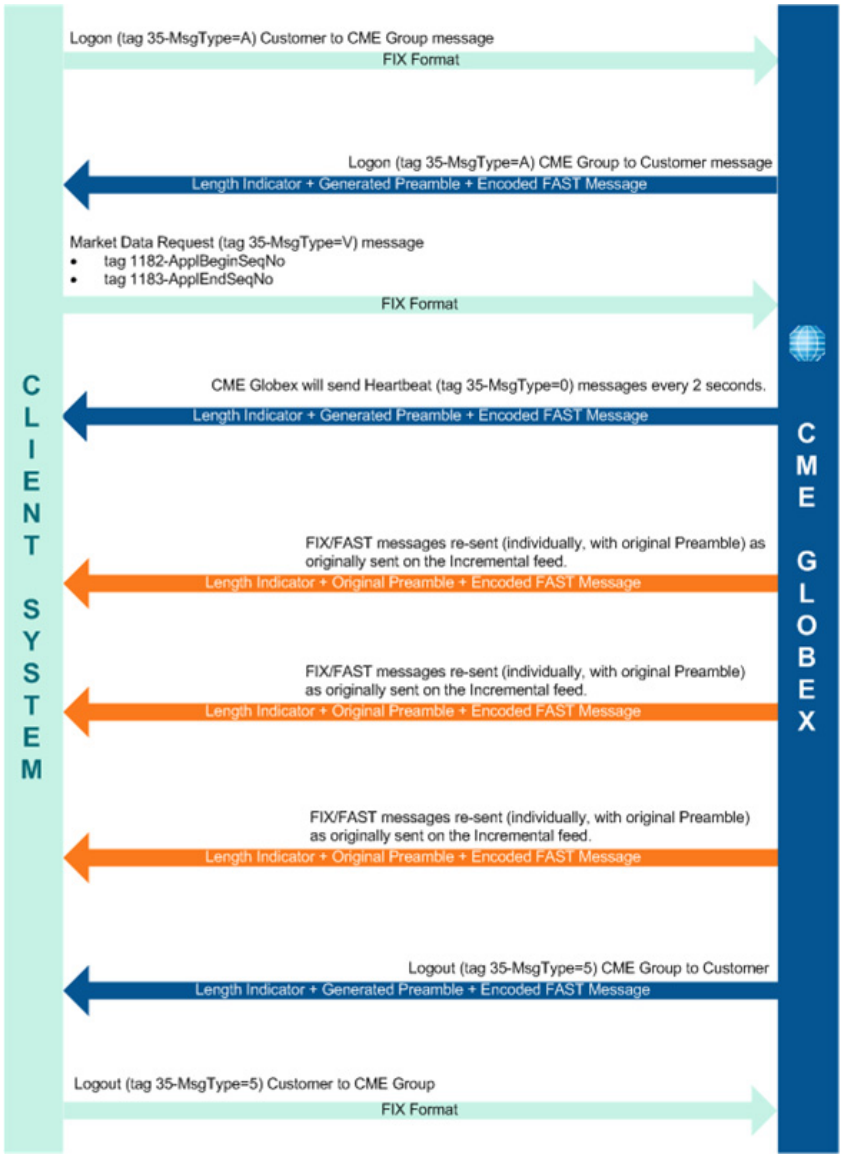
Note: **Client systems must indicate the channel ID (tag 1180-ApplFeedID) and sequence numbers (tag 1182-ApplBeginSeqNo and tag 1183-ApplEndSeqNo) in the Market Data Request (tag 35-MsgType=V) message.**

5. CME Group sends Heartbeat (tag35-MsgType=0) messages to customer every 2 seconds until the first recovery message is sent.
6. CME Group sends FIX/FAST recovery messages that were requested by the customer in the Market Data Request (tag35-MsgType=V) message.
7. CME Group sends a Logout (tag35-MsgType=5) message to the customer.

Note: **CME Group will close the TCP connection if the customer does not send a Logout (tag 35-MsgType=5) message within 5 seconds from the time CME Group sends a Logout (tag35-MsgType=5) message.**

8. Customer sends a Logout (tag 35-MsgType=5) message to CME Group.
9. CME Group closes the TCP connection.

Note: **The username and password for the TCP Replay component for the New Release and Certification testing environments are both "CME" (do not include quotes). The username and password for the production environment are provided after certification is completed. For more information, contact your Global Account Manager.**



8.2 Using the Incremental Market Data Feed to Determine State

In addition to the recovery methods that are offered by CME Group, there are additional mechanisms that client systems can utilize to enhance the recovery process.

- “Instrument Level Sequencing” on Page 108
- “Natural Refresh” on Page 109

8.2.1 Instrument Level Sequencing

Market Data Incremental Refresh messages (tag 35=X) contain instrument sequence numbers (tag 83-RptSeq), in addition to message sequence numbers (tag 34-MsgSeqNum). Every repeating group

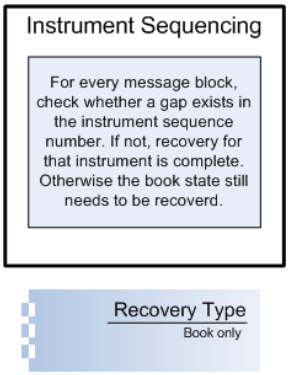
instance of a market data entry contains an incrementing sequence number (tag 83-RptSeq) that is associated with the instrument for which data is present in the block.

Note: **Customers must support Market Recovery functionality.**

Client systems can keep track of the instrument sequence number (tag 83-RptSeq) for every instrument by inspecting incoming data and determining whether there is a gap in the instrument sequence number (tag 83-RptSeq).

- If there is a gap in the instrument sequence number (tag 83-RptSeq), it indicates that data was missed for the instrument when message loss occurred.
- If there is no gap, the data can be used immediately, and it also indicates that the book for this instrument still has a correct, current state.

It is likely that if only a small number of messages have been missed, there will be data in subsequent messages which are not affected by the missing data. If there are 10 instruments in a channel, for example, and the missed messages contain data for 2 of these instruments, any subsequent messages containing data about the other instruments are still valid.



8.2.2 Natural Refresh

The client system must track the state of the book at all times with the FIX Market Data Incremental Refresh message (tag 35-MsgType=X) book update data blocks. It is possible, though not guaranteed, that a set of these book update data blocks can be used to construct the current, correct state of a book without prior book state knowledge. While client systems wait for the recovery of missing data, they may opt to also use a natural refresh algorithm to recover book state prior to recovering the lost data. Natural refresh can also be used to re-instate the top-of-book. Natural refresh should be used for multiple depth book updates only in conjunction with market recovery. Prior to beginning a natural refresh, the entire book should be emptied. Natural refresh assumes no prior knowledge of book state.

The following example illustrates how natural refresh occurs. Note that in this example, using instructions 4 and 5, the top two levels of the offer book are known to be good. This portion of the book can then be displayed or used.

Note: **Natural Refresh is not guaranteed and should not be considered a definitive substitute for recovering lost data. Natural Refresh should only be used in conjunction with Market Recovery.**

Example:

In this example, the book should be emptied:

Top of Book - Best Bid/Ask

Bid		Ask	
Quantity	**Price**	**Price**	**Quantity**

Consider the following received book update data blocks for the offers for a given instrument with a 5-deep book.

1. Insert at book level 3, price 50, quantity 10:
   - tag 279-MDUpdateAction = 0 (new)
   - tag 1023-MDPriceLevel = 3
   - tag 269-MDEntryType = 1
   - tag 271-MDEntrySize = 10
   - tag 270-MDEntryPx = 50

Top of Book - Best Bid/Ask

Bid		Ask	
Quantity	**Price**	**Price**	**Quantity**
		50	10

2. Insert at book level 3, price 40, quantity 10:
   - tag 279-MDUpdateAction = 0 (new)
   - tag 1023-MDPriceLevel = 3
   - tag 269-MDEntryType = 1
   - tag 271-MDEntrySize = 10

- tag 270-MDEntryPx = 40

Top of Book - Best Bid/Ask

Bid		Ask	
Quantity	**Price**	**Price**	**Quantity**
		40	10
		50	10

3. Insert at book level 3, price 30, quantity 10:
   - tag 279-MDUpdateAction = 0 (new)
   - tag 1023-MDPriceLevel = 3
   - tag 269-MDEntryType = 1
   - tag 271-MDEntrySize = 10
   - tag 270-MDEntryPx = 30

Top of Book - Best Bid/Ask

Bid		Ask	
Quantity	**Price**	**Price**	**Quantity**
		30	10
		40	10
		50	10

4. Insert at book level 1, price 10, quantity 10:
   - tag 279-MDUpdateAction = 0 (new)
   - tag 1023-MDPriceLevel = 1
   - tag 269-MDEntryType = 1
   - tag 271-MDEntrySize = 10
   - tag 270-MDEntryPx = 10

Top of Book - Best Bid/Ask

Bid		Ask	
Quantity	**Price**	**Price**	**Quantity**

Top of Book - Best Bid/Ask

Bid		Ask	
		10	10
		30	10
		40	10

5. Update book level 2, price 20, quantity 10:
   - tag 279-MDUpdateAction = 1 (change)
   - tag 1023-MDPriceLevel = 2
   - tag 269-MDEntryType = 1
   - tag 271-MDEntrySize = 10
   - tag 270-MDEntryPx = 20

Top of Book - Best Bid/Ask

Bid		Ask	
Quantity	**Price**	**Price**	**Quantity**
		10	10
		20	10
		30	10
		40	10

8.3 Order Book Reset

Order Book Reset provides a process for synchronizing order books in the unlikely event of a CME Group dual component failure. In this scenario, order books on the channel may be “corrupted”.

Note: **CME Globex leverages the new Order Book Reset method for all markets except NYMEX and COMEX futures and CME and CBOT Interest Rate futures. Refer to “Order Book Reset - Legacy” on Page 115 for more information.**

A new valid tag 269-MDEntryType=J value will be added to the Market Data Incremental Refresh (tag 35-MsgType=X) message to identify an Order Book Reset occurrence.

Table 8.1. Market Data Incremental Refresh (tag 35-MsgType=X) Message

Tag	FIX Name	Format	Valid Values	Description
269	MDEntryType	Char (1)	0 = Bid 1 = Offer 2 = Trade 4 = Opening Price 5 = Closing Price 6 = Settlement Price 7 = Trading Session High Price 8 = Trading Session Low Price E = Simulated Sell F = Simulated Buy **J = Order Book Reset** M = Prior N = Session High Bid O = Session Low Offer	Type of Market Data entry.

The steps to detect this condition and recover are shown below:

6. The Market Data Incremental Refresh (tag 35-MsgType=X) message, Order Book Reset data block is sent down the Incremental feed with tag 269-MDEntryType=J to indicate that there has been a dual component failure and the order books on the channel are corrupted.
7. The client system must empty all order books for the impacted channel.

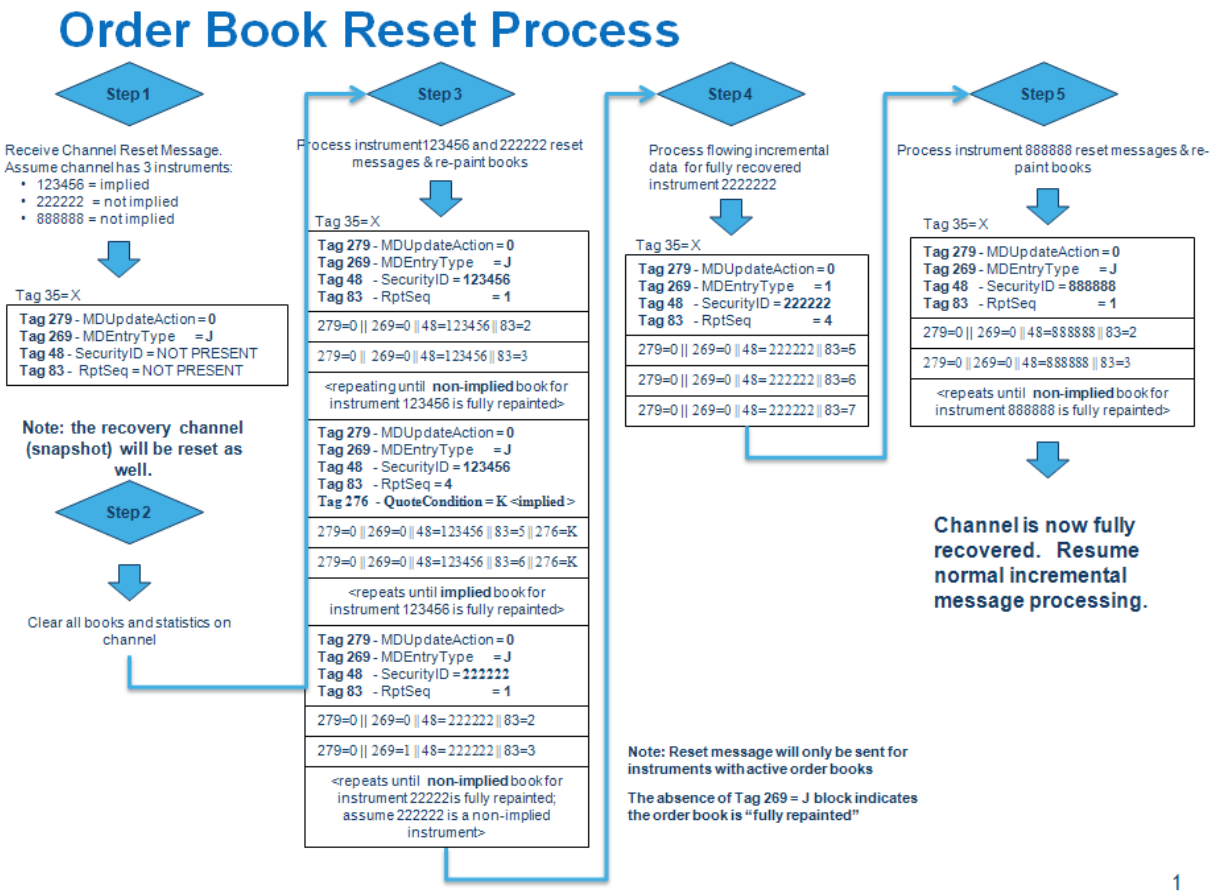
Note: **When the Order Book Reset data block is received, no order entry actions should be taken based on the data found in the order book. The order book and all statistics for the instrument should be emptied on the client system.**

8. The Market Data Snapshot Full Refresh (tag 35-MsgType=W) message on the Market Recovery feed will be deleted for impacted instruments.
9. Market Data Incremental Refresh (tag 35-MsgType=X) messages will be sent to populate all price levels in the order books.

- The first data block will contain tag 269-MDEntryType=J and tag 48-SecurityID, followed by data blocks for each level of the order book as identified in tag 48-SecurityID.
- Client systems should leverage tag 269-MDEntryType=J in combination with tag 48-SecurityID to determine that a book for the instrument identified in tag 48-SecurityID has been recovered.

10. Once a book has been recovered for an instrument (identified by tag 48-SecurityID), CME Globex will disseminate incremental real-time market data for that instrument, and client systems can process data for that instrument. At this point, other instruments on the channel may still be going through the recovery process.

Note: **Tag 83-RptSeq is reset to 1 for each instrument repeating group.**



8.3.1 Order Book Reset - Legacy

The previous Order Book Reset method will continue to be used for NYMEX and COMEX futures and CME and CBOT Interest Rate futures.

For NYMEX and COMEX futures and CME and CBOT Interest Rate futures, CME Globex does NOT send the value J in tag 269-MDEntryType to identify an Order Book Reset. The only indication that market data for an instrument is being resent is that tag 83-RptSeq will be reset to 1 for an instrument repeating group.

If the order books on a channel are corrupted, take the following steps to detect this condition and recover:

1. The Market Data Snapshot Full Refresh (tag 35-MsgType=W) message on the Market Recovery feed are deleted for impacted instruments.
2. Market Data Incremental Refresh (tag 35-MsgType=X) messages are sent to populate all price levels in the order books.
   - The first data block contains tag 48-SecurityID, followed by data blocks for each level of the order book as identified in tag 48-SecurityID.

Note: **Tag 83-RptSeq is reset to 1 for each instrument repeating group.**

3. Once a book has been recovered for an instrument (identified by tag 48-SecurityID), CME Globex will disseminate incremental real-time market data for that instrument, and client systems can process data for that instrument. At this point, other instruments on the channel may still be going through the recovery process.

Revision History

Version	Date Updated	Author	Description
1.0	9/14/07	LM	Initial Release
2.0	8/6/10	DT/AB	• Numerous revisions and corrections. • Incorporated all available client impact documents.
2.1	9/9/10	DT	• Renamed section to “Last Best Price” on Page 64 instead of Last Trade. • Updated description of when the reported trade price is an assigned price by CME Globex in “Trade” on Page 64. • Updated first paragraph in “Example 2 - Two Spread Orders Trading (Not Implied)” on Page 66.