

# THOMSON REUTERS SIMPLE BINARY ENCODING (SBE)

## MARKET DATA INTERFACE SPECIFICATION



© Thomson Reuters 2016. All Rights Reserved.

Thomson Reuters, by publishing this document, does not guarantee that any information contained herein is and will remain accurate or that use of the information will ensure correct and faultless operation of the relevant service or equipment. Thomson Reuters, its agents and employees, shall not be held liable to or through any user for any loss or damage whatsoever resulting from reliance on the information contained herein.

This document contains information proprietary to Thomson Reuters and may not be reproduced, disclosed, or used in whole or part without the express written permission of Thomson Reuters.

Any Software, including but not limited to, the code, screen, structure, sequence, and organization thereof, and Documentation are protected by national copyright laws and international treaty provisions. This manual is subject to U.S. and other national export regulations.

Nothing in this document is intended, nor does it, alter the legal obligations, responsibilities or relationship between yourself and Thomson Reuters as set out in the contract existing between us.

# Contents

<b>About this document.....</b>	<b>5</b>
Intended readership.....	5
In this document .....	5
Feedback.....	5
<b>Chapter 1 Introduction.....</b>	<b>6</b>
1.1 Related Documentation .....	6
<b>Chapter 2 Market Data Feeds Concepts .....</b>	<b>7</b>
2.1 Conflated Market Data Snapshot Full Refresh Feeds .....	7
2.2 Conflated Market Data Incremental Refresh Feeds .....	7
2.3 Conflated Trades Feeds .....	7
2.4 Security Definition Feed.....	8
2.5 Statistics Feed.....	8
<b>Chapter 3 Thomson Reuters Market Data Dissemination &amp; Connectivity.....</b>	<b>9</b>
3.1 Conflated Market Data Dissemination .....	9
3.2 Conflated Market Data Dissemination .....	9
3.2.1 Schema Distribution.....	10
3.2.2 Handling future schema changes/versioning .....	11
3.3 Connectivity Details.....	11
<b>Chapter 4 Simple Binary Encoding Concepts for dissemination of the Market Data and Instruments' Security Definition.....</b>	<b>12</b>
4.1.1 SBE Optimization.....	13
<b>Chapter 5 Conflated Market Data Messaging.....</b>	<b>14</b>
5.1 Packet and Message Headers .....	14
5.1.1 Message Structure.....	14
5.1.2 Packet and Message Headers.....	14
5.1.3 Grouping multiple messages in a single and multiple packets .....	16
<b>Chapter 6 Message Specification.....</b>	<b>18</b>
6.1 Binary Packet Header .....	18
6.2 Message Header .....	18
6.3 Message Header.....	18
6.4 Heartbeat .....	19
6.5 Market Data Snapshot Full Refresh [type 'W'] <MktDataFull> .....	19
6.6 Market data Incremental Refresh Feed .....	20
6.7 Trades Feed - MarketDataIncrementalRefreshtrades [type 'X'] <MktDataInc> .....	21

6.8	Security Definition .....	22
6.9	Statistics Feed.....	25
<b>Chapter 7</b>	<b>Book Management .....</b>	<b>28</b>
<b>Chapter 8</b>	<b>Managing ‘Late Joiners’ .....</b>	<b>31</b>
<b>Chapter 9</b>	<b>Pre-Opening Startup.....</b>	<b>32</b>
<b>Chapter 10</b>	<b>Data Dictionary.....</b>	<b>33</b>

## About this document

### Intended readership

This document provides guidance to client technical teams intending to implement the Thomson Reuters single venue unscreened SBE formatted market data feed service.

### In this document

This document is the specification for a Thomson Reuters single venue unscreened SBE formatted market data feed service.

### Feedback

If you have any comments on this document please contact the [Thomson Reuters TPG Documentation team](#).

# Chapter 1 Introduction

This document is the specification for a Thomson Reuters single venue unscreened SBE formatted market data feed service. The service can provide access to multiple venues, but each venue will have a unique set of addresses and channels that a client of the service will need to connect to for each underlying venue that is supported by the service.

There are 5 feeds that each venue may support:

1. Snapshot Market Data Feed
2. Incremental Refresh Market Data Feed
3. A done Trades Feed
4. Security Definition Feed
5. Statistics Feed /Statistics Feed is out of the scope in the Phase 1 Release/

The SBE protocol is based on the CME Group Market Data Platform (MDP) 3.0 that supports FIX Protocol Simple Binary Encoding concepts and guidance.

The clients will be able to receive the following encoded messages via dedicated feeds and channels:

- The unscreened *MarketDataSnapshotFullRefresh* [type 'W'] **<MktDataFull>** messages
- The unscreened *MarketDataIncrementalRefresh* [type 'X'] **<MktDataInc>** messages
- The *MarketDataIncrementalRefreshtrades* [type 'X'] **<MktDataInc>** messages [Tag 269 =2] with the sets of the trades that occurred within conflated intervals [on day one only the last trade will be published]
- The *SecurityDefinition* [type 'd'] **<SecDef>** messages that publish the Security Definition of all supported instruments
- The Statistics via *MarketDataIncrementalRefresh* [type 'X'] **<MktDataInc>** messages that delivers statistics related to the trading activities on the Thomson Reuters venue(s)

This market data service is designed to support both Level 2 and Level 3 market data feeds. It is also designed to support both conflated and real-time feeds.

Initially, the focus of the document is on the offering of an unscreened, conflated, level 2, multi-cast, market data feed for FX SPOT instruments traded on Thomson Reuters Matching [TRM].

The addition of other Thomson Reuters venues that also will offer an unscreened, conflated, level 2, multi-cast, market data feed for FX SPOT instruments will be done in the same manner but on an additional set of feeds.

The addition of other Thomson Reuters venues that will offer an unscreened, real-time and/or Level 3, multi-cast, market data feed will use the same message types and tags but the message content will reflect the characteristics of this type of feed and will also be done on an additional set of feeds.

## 1.1 Related Documentation

The reader should refer to the following documents for a complete description of the FIX and SBE protocols:

- FIX Protocol Limited, "Financial Information Exchange Protocol (FIX)", Version 5.0SP2.
- <http://www.cmegroup.com/confluence/display/EPICSANDBOX/CME+Market+Data>
- [ftp://ftp.cmegroup.com/SBEFix/Cert/Templates/templates\\_FixBinary.xml](ftp://ftp.cmegroup.com/SBEFix/Cert/Templates/templates_FixBinary.xml)
- <ftp://ftp.cmegroup.com/SBEFix/Cert/Configuration/config.xml>

## Chapter 2 Market Data Feeds Concepts

### 2.1 Conflated Market Data Snapshot Full Refresh Feeds

There are two Conflated UDP Market Data Snapshot Full Refresh Feeds - Feed A and Feed B that deliver the state of the Book in conflated intervals that is Aggregated-By-Price to the predefined depth of book. The Conflated UDP Snapshot Full Refresh Market Data Feeds are based on the encoded packets containing FIX *MarketDataSnapshotFullRefresh* [type 'W'] **<MktDataFull>** messages.

- Feed B is used as a backup in the event that Feed A becomes inoperable; Feed B can also be used for the arbitration if there is a gap of messages on the Feed A.
- The published market data is unscreened meaning that published prices are not checked for credit and as such they could involve orders/market best prices which cannot be matched by a particular counterparty.
- The Market Data Feed updates and publishes the prices/orders at pre-defined fixed intervals only. That means the data that is published to the clients is conflated over a period of time rather than being published based on the Incremental or real-time updates.
- The published prices reflect the current state of the Book defined at the fixed conflation intervals.
- The published prices at each price level are aggregated and represent an 'Aggregation-By-Price' Level 2 view.

### 2.2 Conflated Market Data Incremental Refresh Feeds

There are two Conflated UDP Market Data Incremental Refresh Feeds - Feed A and Feed B that deliver the changes to the Book that occurred between the conflation intervals. The Conflated UDP Incremental Refresh Market Data Feeds are based on the encoded packets containing FIX *MarketDataIncrementalRefresh* [type 'X'] **<MktDataInc>** messages.

- Feed B is used as a backup in the event that Feed A becomes inoperable; Feed B can also be used for the arbitration if there is a gap of messages on the Feed A.
- Each *MarketDataIncrementalRefresh* message (aka delta) will contain the set of market data actions that must be applied in order to update the current state of the Level 2 Book so that it accurately reflects the available liquidity on the venue. Please note that Depth-Of-Book can be different per instrument.
- Each instrument can be configured to its own conflation interval. The conflation interval for each instrument is published in the security definition feed.

### 2.3 Conflated Trades Feeds

There are two Conflated UDP Trades Feeds - Feed A and Feed B that deliver details of trades that occurred within during the incremental conflation interval. The conflated UDP Trades Feeds are based on the encoded packets containing the FIX *MarketDataIncrementalRefresh* [type 'X'] **<MktDataInc>** messages having Tag 269 set to 2.

- Feed B is used as a backup in the event that Feed A becomes inoperable; Feed B can also be used for the arbitration if there is a gap of messages on the Feed A.

## 2.4 Security Definition Feed

There are two Conflated UDP Trades Feeds - Feed A and Feed B that delivers an instruments Security Definition details using encoded packets containing the *SecurityDefinition* [type 'd'] **<SecDef>** messages

- Feed B is used as a backup in the event that Feed A becomes inoperable; Feed B can also be used for the arbitration if there is a gap of messages on the Feed A.

## 2.5 Statistics Feed

There are two Conflated Statistics Feeds - Feed A and Feed B that delivers statistics related to the market events using encoded packets based on the FIX *MarketDataIncrementalRefresh* [type 'X'] **<MktDataInc>** messages. These are not used to update the Book

- Feed B is used as a backup in the event that Feed A becomes inoperable; Feed B can also be used for the arbitration if there is a gap of messages on the Feed A.

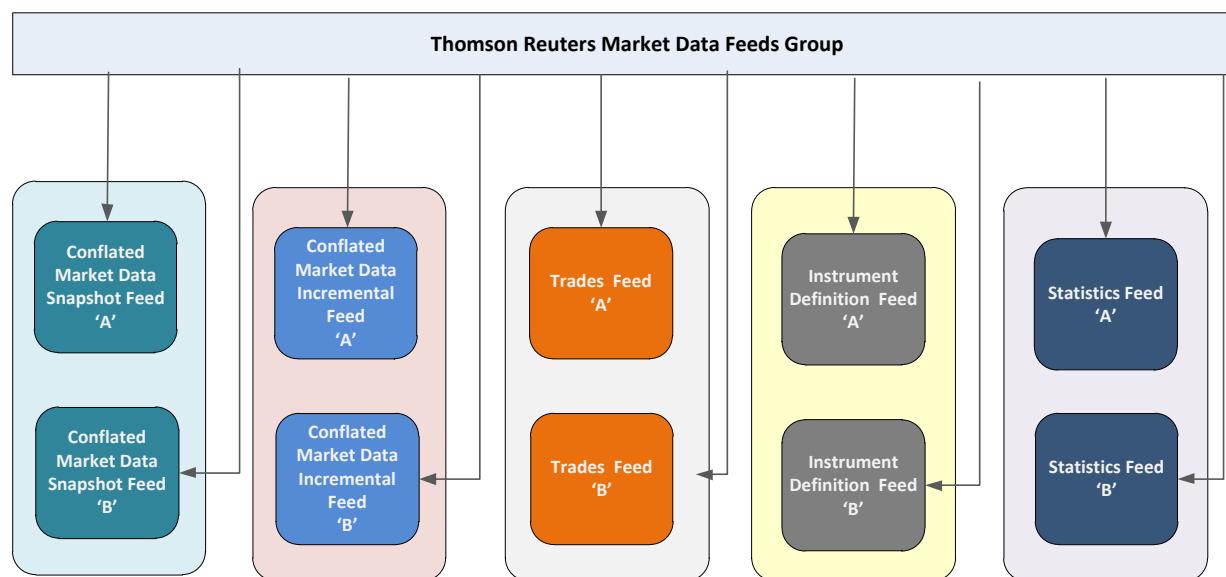


## Chapter 3 Thomson Reuters Market Data Dissemination & Connectivity

### 3.1 Conflated Market Data Dissemination

Based on the concepts presented above, Market Data dissemination is organized by a Market Data Group [MDG] as it's shown below.

MDG is a set of feeds and channels that provide all the market data provided by a particular venue.



The conflated market data snapshot full refresh feeds (both A and B) publish at intervals independent of the market data incremental refresh conflation intervals of each individual instruments. Each snapshot contains the complete currently visible state of an individual instrument's book.

The trades feeds (both A and B) publish at intervals independent of the market data incremental refresh conflation intervals of each individual instruments.

- Feeds B, as is stated above, are used for the backup and arbitration purposes in the cases feeds A became unavailable or if a packet is not received on either feed A or feed B. Schema Dissemination

### 3.2 Conflated Market Data Dissemination

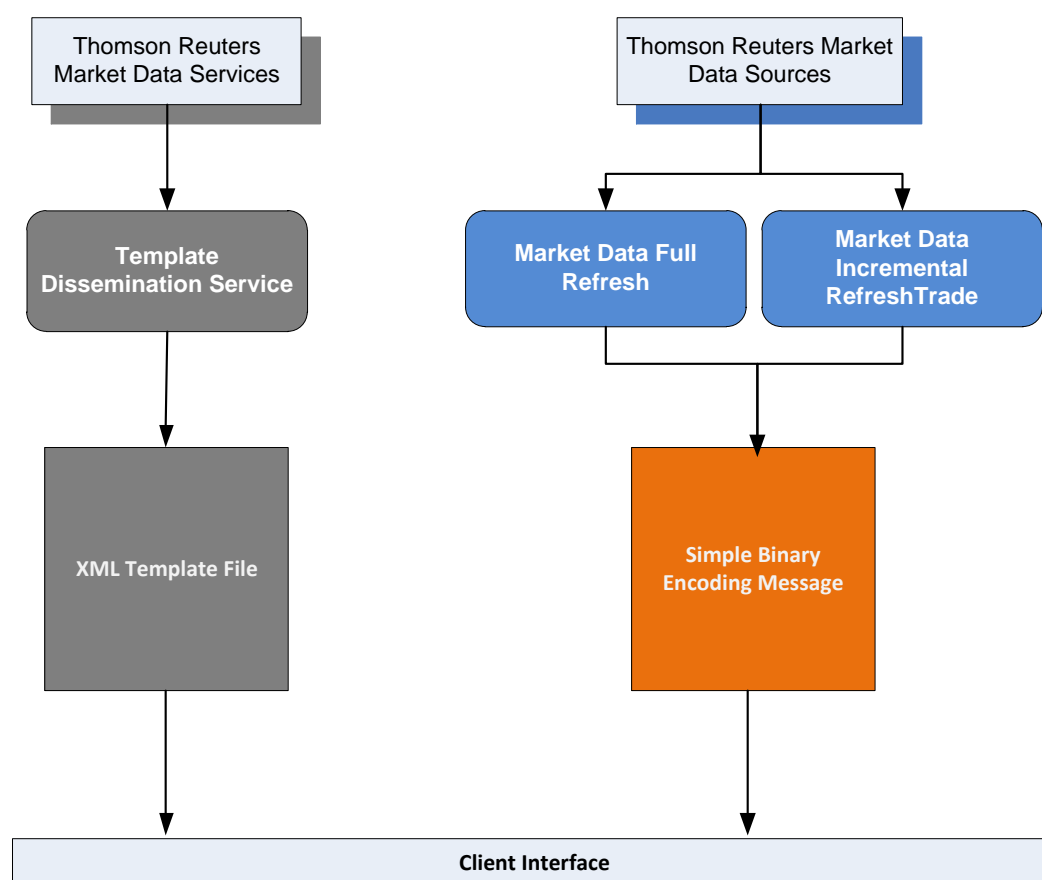
This is the method for client systems to receive all active Thomson Reuters templates.

Thomson Reuters provides single XML schema file that contains FIX templates for all message types sent by Thomson Reuters.

- A message schema is composed of metadata that specifies the components of a message and their data types and identifiers. For Simple Binary Encoding, message schemas are expressed as an XML template.
- A FIX 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 Thomson Reuters and client systems in XML syntax using the Simple Binary Encoding XML schema.

- Each template has a unique Template ID that describes the format of the binary encoded message.
- A Template ID is present in every message to provide a reference to the correct template.
- The Template ID is unique and is included in every message header, allowing the client system to apply the correct schema to the message upon receiving it
- The templates.xml file should be versioned each time an update is made. All elements in a message schema are of the same version. The first version of a schema is version zero, and the version number is incremented each time a schema is changed. The “sinceVersion” attribute indicates when a template is extended.

The following diagram shows how the Thomson Reuters Market Data service provides the message templates required by the client system to process the Incremental market data feed. For the illustration purpose some other utility feeds, like Statistics and Security Definition, are not shown on this diagram.



### 3.2.1 Schema Distribution

Please note that details regarding XML schema download, Channels' IP Addresses and Ports will be communicated by Thomson Reuters later.

### 3.2.2 Handling future schema changes/versioning

Future schema/versioning changes, if require, will be managed in the following way:

- Additional field(s) on an existing SBE message
  - A new single field will be placed at the end of the non-repeating part of the FIX message
  - A new field in the repeating group will be placed at the end of the repeating group
- New Asset Class or new Venue
  - A new schema with new message IDs will be added for each new Asset Class and a new Venue. The new schema will reflect the specifics in each of the newly defined messages on the SBE feed
- Thomson Reuters will notify clients' in advance about upcoming changes and will provide a time window to apply these changes.
  - All the changes [new schema] initially will be available in the UAT environment. Clients who decide to apply these changes will be require to confirm their compliance with the new schema changes / versioning before 'going live' in Production environment.

## 3.3 Connectivity Details

Please refer to the Thomson Reuters FX SBE Market Data Connectivity Guide for specific details for connecting to this service. It contains the information necessary for client integration and production connection details for all feeds, channels, and any other services provided for specific venues.

## Chapter 4 Simple Binary Encoding Concepts for dissemination of the Market Data and Instruments' Security Definition.

As it is stated above, the Simple Binary Encoding (SBE) is used for the dissemination of the Market Data and Instruments' Security Definition.

- Simple Binary Encoding (SBE) is based on a simple primitive encoding, and is optimized for low bandwidth, low latency, and direct data access.
- SBE provides:
  - Independence between number of events, messages, and packets, including:
    - Multiple messages per packet
    - A single event over multiple packets. Each packet contains a complete message as defined by the FIX specification, which allows client systems to start processing the message once the first packet is received.
    - Fixed-length fields, which allow direct data access to fields in the message based on offsets and eliminate the need to parse entire messages.
- The Binary Type System offers the following benefits:
  - Provides a means to specify precision of decimal numbers and timestamps, as well as valid ranges of numbers
  - Provides a consistent system of enumerations, Boolean switches and multiple-choice fields.
- Simple Binary Encoding supports traditional FIX field types without conversions. These became possible by:
  - Usage of native binary data types and simple types derived from native binaries, such as prices and timestamps.
  - Preference for fixed positions and fixed length fields, supporting direct access to data and avoiding the need for management of variable-length elements which must be sequentially processed.
  - Decimal numbers for prices have flexible or fixed precision
  - Timestamps, times and dates are sent as numeric units of time rather than verbose strings
- SBE supports all FIX semantics.

### 4.1.1 SBE Optimization

SBE has two features to control alignment of message elements.

- The length of a block is controlled with the **blockLength** attribute of a message or group. When applied to a message, it controls the length of the root level of the message, prior to any repeating groups. When applied to a group, it controls the length of each entry of the repeating group. The **blockLength** attribute is not required. By default, the length of a block is the sum of its field sizes. When specified, it must be at least that much, but can be greater. When greater than the field lengths, the extra space resides at the end of the block.
- The position of an individual field can be controlled with the **offset** attribute. Since individual field alignment is not considered crucial, this field attribute is optional. When not specified, the field is packed to the previous field without padding.

## Chapter 5 Conflated Market Data Messaging

Current offering of the Conflated Market Data messages is based on the following events:

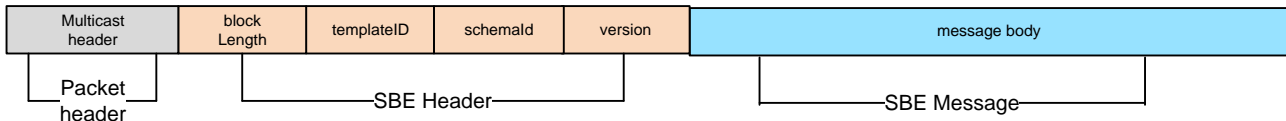
- At fixed intervals using *MarketDataSnapshotFullRefresh* [type 'W'] messages to reflect the current state of the Market Data Book
- At fixed intervals using *MarketDataIncrementalRefresh* [type 'X'] messages that need to be actioned to update to the current state of the Market Data Book.
- At fixed intervals using *MarketDataIncrementalRefreshtrades* [type 'X'] messages to communicate trade events that occurred within an incremental refresh conflation interval.
- Market Data Incremental Refresh messages will be grouped by conflation interval into packets. The order of currency pairs in each packet are in a pre-configured order: this will allow clients to determine quickly if there is an update for a particular instrument or not.

### 5.1 Packet and Message Headers

#### 5.1.1 Message Structure

The encoded FIX transmission is sent in a packet structured as follows:

- Packet header - contains packet sequence number, sending time.
- Message header - contains block length, Template ID, Schema ID, and Version.
- FIX header - indicates FIX message type (example: 35=X)
- FIX message body - event driven business data such as Book updates and trade summary.



#### 5.1.2 Packet and Message Headers

Each packet sent on all feeds should contain a packet sequence number and a timestamp. The packet number is unique to each packet sent and each channel has its own separate set of sequence numbers that increment sequentially with each packet and are reset weekly. In a case of the intra week restart of the Market data server, the sequence numbers will restart with a higher number, and client will need to follow the steps described in Section 8

Packets may contain a single or multiple messages.

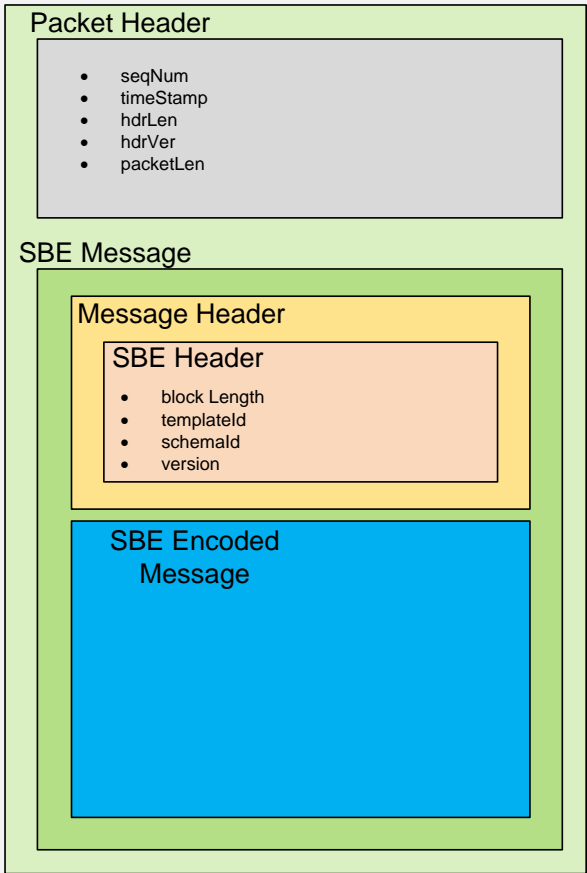
- Each packet is **prefixed** with:
  - Sequence Number
  - Sending Time
  - (HdrLen
  - HdrVer
  - PacketLen
- Each message is **prefixed** with:
  - Template ID

- SchemaID
- Version

Packets can be organized in one of the following ways:

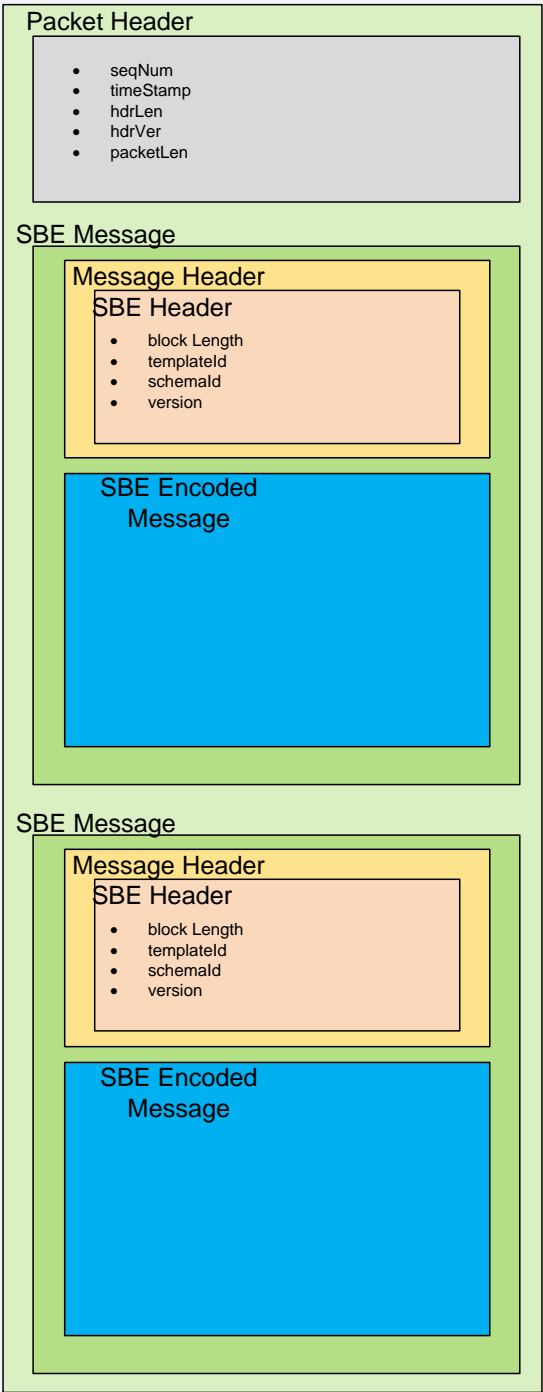
**Packet with Single Message**

A packet with a single message includes only one message within packet.  
The structure of a packet with a single message is shown below.



**Packet with Multiple Messages**

This model supports concept of packaging multiple messages within a single packet.  
The structure of a packet with a multiple messages is shown below



### 5.1.3 Grouping multiple messages in a single and multiple packets

Grouping of multiple messages within a single packet or spanning them across multiple packets is not predictable and is driven by the size of the messages and the standard size of the packet.

- Instrument priority ordering within messages
  - Instrument priority ordering consistently applies on all messages across the all four channels.
  - Instrument priority ordering is always maintained for the messages which are sent on any given channel at any predefined conflation point, for example:
    - The assigned priority order for the following instruments is as following: AUD/USD, USD/CHF, GBP/USD, USD/JPY, and USD/CAD.
    - For AUD/USD, GBP/USD, USD/CAD the conflation interval is 50 ms while for USD/CHF and USD/JPY the conflation interval is 100 ms.
    - At 50ms conflation MktDataInc messages are sent: AUD/USD, GBP/USD, USD/CAD if they all have an update to be sent
    - At 50ms conflation MktDataInc messages are not sent if there are no updates to any of the currency pair [e.g. AUD/USD, GBP/USD, USD/CAD]
    - At 50ms conflation MktDataInc messages are sent: AUD/USD, GBP/USD [if there is no updates to USD/CAD]
    - At 100 ms conflation MktDataInc messages are sent: AUD/USD, USD/CHF, GBP/USD, USD/JPY, and USD/CAD if they all have an update to be sent.
  - The Instrument set, and priority order will only be change with prior warning and only at weekends.
  - Clients can determine the priority order based on the order in which SecurityDefinition (or MktDataFull) messages are received.
- Grouping multiple messages in a single packet
  - MDS will group multiple FIX messages within a single packet applying the following rules:
    - The size of the messages are small enough to fit multiple full messages in one packet
    - Grouping is within a single channel (i.e. one of MktDataFull, MktDataInc, MDTrades, SecurityDefinition)
    - Grouping is applicable to the different instruments/currency pairs and can be used for any of the messages within a single channel

- Large messages spanning across multiple packets

MDS multicast will span packets applying the following approach:

- Split the required data into multiple FIX messages
  - The fields outside of the repeating group will be present in each message
  - The repeating group will be split between the messages
    - Example: Message has to span across 2 packets:
      - 1<sup>st</sup> packet contains the non-repeating set of the fields and the first set of 'X' repeating groups.
      - 2<sup>nd</sup> packet contains the same set non-repeating fields and the remaining set of 'Y' repeating groups.
    - These packets should be treated by the customers in the 'cumulative' manner as they are sent in a single conflation interval before it ends.



- If the remaining portion of the packet permits, a new message can be included in that packet.
- If the size of the packet permits, the next message can be fit in the packet

## Chapter 6 Message Specification

This section provides the messages layout for each supported FIX message type.

### 6.1 Binary Packet Header

A standard header should be included in all packets sent on Real-Time Feed, Snapshot and Instrument Recovery Feeds.

Name	Binary Type (Size)	Number of Bytes	Offset	Req' d	Description
<b>Start of the Binary Packet Header</b> –standard technical header included in all packets sent on the Snapshots					
MsgSeqNum	uint64	8		Y	
SendingTime	uint64	8		Y	
hdrLen	uint8	1		Y	20
hdrVer	uint8	1		Y	1
packetLen	uint16	2		Y	
<b>End of the Binary Packet Header</b>					

### 6.2 Message Header

Each message in the packet starts with a Binary message header that consists of the Binary Size and SBE header (Length, TemplateID, SchemaID, and Version).

Name	Binary Type (Size)	Number of Bytes	Offset	Req' d	Description
<b>Start of the Message Header –</b>					
<b>Simple Binary Encoding Header</b>					
BlockLength	uint16	2		Y	Length of root of the message
TemplateID	uint16	2		Y	Template ID used to encode the message
SchemaID	uint16	2		Y	ID of the system publishing the message
Version	uint16	2		Y	Schema version
<b>End of the Message Header</b>					

### 6.3 Message Header

Name	Binary Type (Size)	Number of Bytes	Offset	Req' d	Description
MsgType				Y	Defines message type

## 6.4 Heartbeat

Heartbeat (tag 35-MessageType=0) message is sent on all the feeds in periods of no activity at a configurable time interval and consists of only a standard technical header and a FIX message header.

## 6.5 Market Data Snapshot Full Refresh [type 'W'] <MktDataFull>

As is stated above, the *MarketDataSnapshotFullRefresh* will deliver the state of the Book in conflated intervals that is Aggregated-By-Price to the predefined Book depth.

In addition, the following applies:

- Conflation interval is specific to the instrument and is defined in the Security Definition message.
- Depth Of Book (**Depth of Book**) is specific to the instrument and is defined in the Security Definition message.
- In addition to the DOB setting, the DOB can optionally be further limited for a specific instrument. A Price Points from TOB (Top of Book) Threshold can be configured and that threshold may limit DOB to less than the DOB setting for the instrument. This is defined in the Security Definition message.
- The message content of the Market Data Snapshot Full Refresh [type 'W'] <MktDataFull> is shown below:

Rep. Grp	Tag	Field Name	Y/N/C	Type	Description
369		LastMsgSeqNum Processed	Y	SeqNum	Sequence number of the last Incremental feed packet for the instrument being processed. This value is used to synchronize the snapshot loop with the real-time feed and or any other feeds as required by the business flow.
48		SecurityID	Y	String	Unique instrument ID
83		RptSeq	Y	Int	Sequence number of the last Market Data Entry for the instrument reflected in the message
60		TransactTime	Y	UTCTimestamp	Start Transaction Time of the last event instrument participated in.
282		MDEntryOriginator	N	String	Text field indicating the originator of the market data.
264		MarketDepth	Y	Int	Describes the type of Book for which the feed is intended. Supported value: 2 = Book Depth
167		SecurityType	N	String	Indicates type of security. Supported value: FXSPOT <ul style="list-style-type: none"> <li>• If not presented the Security Type is FXSPOT.</li> </ul>
268		NoMDEntries	Y	NumInGroup	Number of entries that follow (Will be 1 or more)

Rep. Grp	Tag	Field Name	Y/N/C	Type	Description
→	269	MDEntryType	Y	Char	Type of Market Data entry. Supported values: 0 = BID 1 = OFFER
→	270	MDEntryPx	Y	Price	Current value of bid/offer for currency pair.
→	271	MDEntrySize	Y	Qty	Current quantity for quoted rate.

## 6.6 Market data Incremental Refresh Feed

The **Market Data IncrementalRefresh Feed** is used to disseminate the changes to the Book that occurred within conflated intervals using the encoded packets containing the binary version of the *MarketDataIncrementalRefresh* [type 'X'] <MktDataInc> messages. The **Market Data IncrementalRefresh Feed** is published via a separate feed channel from the Full Snapshot feed channel.

- The aggregated depth of the Book view is maintained with the following data blocks:
  - Add - create/insert a new price at a specified price level
  - Change - change quantity for a price at a specified price level
  - Delete\* - remove a price at a specified price level
- Each MarketDataIncrementalRefresh message (aka delta) will reflect a 'final' set of the changes that reflect market data actions to be taken in order to update the current state of a full snapshot of the aggregate level 2 Book.
  - If there are no delta changes within conflation interval, then the delta will not be published.
  - For an example refer to below.
- When there is an Add action to create a new price level within the pre-defined DOB, the new price level must be inserted in the Book and any price levels past the pre-defined DOB should be removed.
- When there is a Delete action to remove a price level in the Book, that price level must be removed from the Book and an Add action will accompany the delete so that full DOB can be maintained.
- When there is a Change action to update the displayed available quantity that quantity should be applied as an update to the price level. Price level ads and deletes are never reflected in a change action.

The message content of the **Market Data IncrementalRefresh** [type 'X'] <MktDataInc> is shown below:

REP. GRP	TAG	FIELD NAME	Y/N	TYPE	DESCRIPTION
48		SecurityID	Y	String	Unique instrument ID
83		RptSeq	Y	Int	Sequence number of the last Market Data Entry for the instrument reflected in the message
60		TransactTime	Y	UTCTimestamp	Start Transaction Time of the last event security participated in.

REP. GRP	TAG	FIELD NAME	Y/N	TYPE	DESCRIPTION
282		MDEntryOriginator	N	String	Text field indicating the originator of the market data.
268		NoMDEntries	Y	NumInGroup	Number of entries that follow (Will be 1 or more)
→	279	MDUpdateAction	Y	Char	Type of Market Data updates action. Supported values: 0 = NEW 1 = CHANGE 2 = DELETE
→	269	MDEntryType	Y	Char	Type of Market Data entry. Supported values: 0 = BID 1 = OFFER
→	270	MDEntryPx	Y	Price	Current value of bid/offer for currency pair.
→	271	MDEntrySize	Y	Qty	Current quantity for of bid/offer.

## 6.7 Trades Feed - MarketDataIncrementalRefreshtrades [type 'X'] <MktDataInc>

The Trades Feed *MarketDataIncrementalRefreshtrades* [type 'X'] will publish the trade(s) that occurred within each incremental conflation interval.

In addition, the following applies:

- The conflation intervals of the Trades Feed is determined by the conflation interval of the *MarketDataIncrementalRefresh* for each particular instrument, e.g. trades messages will be published in the same interval as the instruments market data conflation interval.
- Each *MarketDataIncrementalRefreshtrades* [type 'X'] message should include the sequence number of the Market Data Incremental Refresh packet for the specific conflation period covered by the trades message. This associates the trades message with the *MarketDataIncrementalRefresh* [type 'X'].
- Each Trade Feed's packet should include details of the trade(s) within the conflation interval between last 2 published *MarketDataIncrementalRefresh* messages
  - These details are:
    - Rate
    - Direction (Paid / Given)
    - Instrument
    - Timestamp for each trade included.
- The *MarketDataIncrementalRefreshtrades* [type 'X'] message will only be published for conflation intervals where trades have occurred.

Below is the content of the *MarketDataIncrementalRefreshtrades* [type 'X'] message to be used with the Trades Feed:

REP. GRP	TAG	FIELD NAME	Y/N	TYPE	DESCRIPTION
48		SecurityID	Y	String	Unique instrument ID
75		TradeDate	Y	LocalMktDate	Used to specify the trading date for which a set of market data applies
282		MDEntryOriginator	N	String	Text field indicating the originator of the market data.
268		NoMDEntries	Y	NumInGroup	Number of entries that follow (Will be 1 or more)
→	279	MDUpdateAction	Y	Char	Supported value: 0 = New
→	269	MDEntryType	Y	Char	Supported value: 2 = Trade
→	60	TransactTime	Y	UTCTimestamp	Timestamp of the last event security participated in, e.g. Timestamp for the trade
→	64	SettlDate	Y	LocalMktDate	Specific date of trade settlement (SettlementDate) in YYYYMMDD format.
→	270	MDEntryPx	Y	Price	Trade price.
→	271	MDEntrySize	Y	Qty	Current quantity for BID/OFFER. The value is always 0 for Phase 1
→	30432	AggressorSide	Y	Char	Paid / Given [from the 'Aggressor' perspective] Supported values: 0 = No aggressor 1 = BUY /Paid/ 2 = SELL /Given/

## 6.8 Security Definition

As it is stated above, the SecurityDefinition Feeds are broadcasting Instruments Security Definitions and are based on the structure of the *SecurityDefinition* [‘type d’] <SecDef> message.

- The SecurityDefinition of each instrument is broadcasted via a dedicated packet
- The SecurityDefinition Feeds are broadcasting Instruments Security Definitions in the predefined instrument priority ordering
- The SecurityID values for the instrument may change between trading weeks, although they are not changing within the trading week.
- Clients must take the SecurityID values from the SecurityDefinition channel at runtime.
- At the start of the trading week (Sunday noon GMT when SecurityDefinition starts to be published) the very first SecurityDefinition message for each instrument will carry the value of ‘A’ [add] in the tag 980 [SecurityUpdateAction], and all the subsequent messages either M or N (M when value date has changed, otherwise N as nothing else can change during week on NMH)
- At the end of the trading week SecurityDefinition data for all instruments must be treated as invalid, and their applications should process the new set of SecurityDefinition data at the start of the next trading week.
- Tag 779 [LastUpdateTime] – represents the timestamp indicating when the MDS application received notification from NMH [New Matching Host] of a change. The tag will change daily due to Value Date (tag 64 [SettlDate]) changes at day rollover, and the timestamp will not be the exact

time at which the rollover occurred on the NMH venue, it will be a short period after (few seconds perhaps).

During the MDS failure recovery the timestamp will be updated to indicate the recovery time (and tag 980 [SecurityUpdateAction] may be set to A), when there has not actually been any change or addition

Below is the content of the SecurityDefinition [type'd'] <SecDef> message

TAG	FIELD NAME	TYPE	DESCRIPTION
980	SecurityUpdate Action	Char	Indicates update action for the given security. Supported values are: <ul style="list-style-type: none"> <li>• A = Add represents newly added instrument</li> <li>• D = Delete represents deletions of a Security Definition for an instrument</li> <li>• M = Modify represents modifications to a Security Definition for an instrument</li> <li>• N = No changes in respect to the previous security definition of the instrument</li> </ul>
779	LastUpdateTime	UTCTime stamp	Timestamp of when the instrument was last added, modified or deleted.
1180	ApplID	String	The channel ID as defined in the XML Configuration file.
282	MDEntryOriginator	String	Text field indicating the originator of the Instrument Security Definition
55	Symbol	String	Symbol, e.g. EUR/USD
48	SecurityID	Int	Unique instrument ID
22	SecurityIDSource	Int	8 = Exchange Symbol. This value is always 8 for Thomson Reuters. This Tag required if tag 48-SecurityID is specified.
167	SecurityType	String	Indicates type of security. Valid values: <ul style="list-style-type: none"> <li>• FXSPOT</li> </ul> If not presented the Security Type is FXSPOT.
64	SettlDate	LocalMkt Date	Instrument's Settlement Date; always expressed in YYYYMMDD format. <ul style="list-style-type: none"> <li>• Does not require for FXSPOT.</li> </ul>
30375	Currency1 [BASECUR]	Currency	ISO Currency code indicating the first currency in the currency pair represented by Symbol. For instance for "EUR/JPY" Currency1 is "EUR".
30376	Currency2	Currency	ISO Currency code indicating the second currency in the currency pair represented by Symbol. For instance for "EUR/JPY" Currency2 is "JPY".
30378	BasisPoint	Int	Integer value representing the decimal position of one point, or pip used to deal in the currency. For instance for EUR/JPY a point (pip) is represented by decimal place 2, whereas for EUR/GBP a pip is represented at decimal place 4.

TAG	FIELD NAME	TYPE	DESCRIPTION
30379	RatePrecision [PIPSIZE]	Int	Integer value representing the highest decimal precision used to show prevailing prices in the currency pair specified by Symbol. For instance for EUR/JPY the rate precision is typically 2, whereas for EUR/GBP the rate precision is typically 5.
30380	RateTerm	Int	ISO Currency code indicating the currency in terms of which the price for the currency pair represented by Symbol is expressed. For instance for "EUR/JPY" RateTerm is "JPY". Supported values: 1 = BASE 2 = QUOTED
30381	Currency1AmtDecimals	Int	Integer value representing the decimal precision with which the quantity of Currency1 will be measured for executions in the currency pair specified by Symbol. This is typically 2.
30382	Currency2AmtDecimals	Int	Integer value representing the decimal precision with which the quantity of Currency2 will be measured for executions in the currency pair specified by Symbol. This is typically 2.
30433	RGTSMDPS		Indicates how many of the rightmost digits of the price to treat as the fractional part of a pip
30434	LEFT_DPS		The maximum number of permitted left decimal places.
30435	RIGHT_DPS		The maximum number of permitted right decimal places.
30436	CLS		Indicates if the instrument is eligible for CLS. 0 = FALSE 1 = TRUE
1143	MaxPriceVariation	float	The maximum price variation of an execution from one event to the next for a given security. <ul style="list-style-type: none"> <li>Expressed in pips</li> </ul>
30437	SnapshotConflation Interval	Time	Conflation interval for broadcasting the instrument's Snapshot Full Refresh messages
30438	IncRefreshConflation Interval	Time	Conflation interval for broadcasting the instrument's Incremental messages
30430	TradesFeedConflation Interval	Time	Conflation interval for broadcasting the instrument's trades messages
30431	SecurityDefinition Conflation Interval	Time	Conflation interval for broadcasting the instrument's Security Definition messages
30439	DepthOfBook	Int	Defines the number of the price levels to be displayed in the Book for the instrument.
562	MinTradeVol	Qty	The minimum trading volume for an instrument that must be traded for the price to be used to change the market high/low. Phase 1: the value is NULL; all the applicable levels will be displayed.



## 6.9 Statistics Feed

As is stated above, the Statistics Feed is broadcasting statistics related to the market events that are not used to update the Book. The Statistics Feed structure is based on of the *MarketDataIncrementalRefresh* [type 'X'] message.

- All the statistics' attributes defined below are per instrument
- Each packet is per instrument and is broadcasted in the conflated intervals
- Conflation Intervals are specific to the instrument.
- Conflation Interval may or may not be synchronized with the respective instrument's incremental Refresh Feed conflation interval.
- Conflation Interval for each of the feed is communicated to the customers via Security Definition Feed.
- Minimum trade volume for the *Daily Market High and Market Low Trade Price* is specific to the instrument and communicated to the customers via Security Definition Feed.
- *VWAP High and Low Trade Price* is specific to the instrument and communicated to the customers via Security Definition Feed.
- High Bid and Low Ask.
  - FIX Syntax for High Bid within **MDIncGrp <Inc>** repeating group:

SE	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data updates action.
269	MDEntryType	N	N = session high bid.
48	SecurityID		Unique instrument ID
270	MDEntryPx		Price of the Market Data Entry

- FIX Syntax for Low Ask within **MDIncGrp <Inc>** repeating group:

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data updates action.
269	MDEntryType	O	O = session low ask.
48	SecurityID		Unique instrument ID
270	MDEntryPx		Price of the Market Data Entry

- If any of orders left on the Book following an opening, then they will be evaluated to generate a better high bid / low ask statistics:
  - If there are orders remaining in the Book from the first pre-open of the day, the top of Book price of each side will be sent as the High Bid / Low Ask respectively.
  - High Bid / Low Ask will not be sent if there are no orders in the Book after the opening event, even in situations where there was an IOP generated or trades occurred at the opening.
- If an Aggressing Order automatically trades and does not rest on the Book, it will not be factored into the High Bid / Low Ask calculations. If an Aggressing Order partially trades and the balance rests on the Book, the order's price will be used to calculate the High Bid or Low Ask.

- *Daily Absolute High and Low Trade Price.* The High Trade Price data block broadcast the highest trade price with the pre-defined per instrument frequency [conflation intervals]. Likewise, the Low Trade Price data block broadcast the lowest trade price with the pre-defined per instrument frequency [conflation intervals].

- FIX Syntax for the Absolute High Trade Price within **MDIncGrp <Inc>** repeating group:

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new.
269	MDEntryType	7	7 = the highest trading price.
48	SecurityID		Unique instrument ID
270	MDEntryPx		Price of the Market Data Entry

- FIX Syntax for the Absolute Low Trade Price within **MDIncGrp <Inc>** repeating group:

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new.
269	MDEntryType	8	8 = lowest trading price.
48	SecurityID		Unique instrument ID
270	MDEntryPx		Price of the Market Data Entry

- *Daily Market High and Low Trade Price.* The Market High Trade Price data block broadcasts the highest trade price only after the predefined minimum trade volume is reached. Likewise, the Market Low Trade Price data block broadcasts the lowest trade price for the instruments only after the predefined minimum trade volume is reached.

- FIX Syntax for the Market Session High Trade Price within **MDIncGrp <Inc>** repeating group:

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data updates action.
269	MDEntryType	7	7 = the high trading price.
48	SecurityID		Unique instrument ID
270	MDEntryPx		Price of the Market Data Entry
1020	TradeVolume		Commutative traded volume

- FIX Syntax for the Session Low Trade Price within **MDIncGrp <Inc>** repeating group:

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data updates action.
269	MDEntryType	8	8 = the low trading price.
48	SecurityID		Unique instrument ID
270	MDEntryPx		Price of the Market Data Entry
1020	TradeVolume		Commutative traded volume

- *VWAP High and Low Trade Price*. The VWAP High Trade Price data block broadcasts the highest VWAP trade price for the instrument within the pre-defined conflation interval. Likewise, the VWAP Low Trade Price data block broadcasts the lowest trade price for the instrument within pre-defined conflation interval.

- FIX Syntax for the VWAP High Trade Price within **MDIncGrp <Inc>** repeating group:

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data updates action.
269	MDEntryType	7	7 = session high trading price.
48	SecurityID		Unique instrument ID
270	MDEntryPx		Price of the Market Data Entry
828	TrdType	6	Weighted Average Price Trade
943	TimeBracket		Time Bracket VWAP is calculated for
1020	TradeVolume		Cumulative traded volume

- FIX Syntax for the VWAP Low Trade Price within **MDIncGrp <Inc>** repeating group:

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data updates action.
269	MDEntryType	8	8 = session low trading price.
48	SecurityID		Unique instrument ID
270	MDEntryPx		Price of the Market Data Entry
828	TrdType	6	Weighted Average Price Trade
943	Time Bracket		Time Bracket VWAP is calculated for
1020	TradeVolume		Cumulative traded volume

## Chapter 7 Book Management

This section provides description how the multiple depths Book can be managed by the clients.

Assumptions:

- Each MarketDataIncrementalRefresh message (aka delta) will reflect a 'final' set of the changes that reflect market data actions to be taken in order to update the current state of a full snapshot of the aggregate level 2 Book.
  - If there are no delta changes within conflation interval, then the delta will not be published.
  - For an example refer to the example below.
- As its stated above, the Aggregate Depth of Book view is maintained with the following data blocks:
  - Add - create/insert a new price at a specified price level
  - Change - change quantity for a price at a specified price level
  - Delete - remove a price at a specified price level

An example below illustrates how the delta incremental Refresh message should look based on the changes occurred within a single conflated interval:

LEVEL 2 AGGREGATED DEPTH BOOK			
BID		ASK	
Quantity	Price	Price	Quantity
5	1.0981	1.0983	10
12	1.0980	1.0984	14
8	1.0979	1.086	12
14	1.0978	1.0987	10
6	1.0975	1.0989	20

- Bid size at the price MD Price Level 1 changed from 5 MIO to 3 MIO
- New best Bid of the size of 5 MIO at the price 1.0982 has been added
- Bid Size at the MD Price Level 1 changed from 5 MIO to 7 MIO
- Bottom 2 Offer price levels has been deleted
- New Offer of the size of 5 MIO at the price 1.0987 has been added to the MD Price Level 4
- New Offer of the size of 1 MIO at the price 1.0990 has been added to the MD Price Level 5
- Offer size at MD Price Level 1 and 2 respectively changed from 10 MIO to 5 MIO, and from 14 MIO to 10 MIO

These changes being summarized look as following:

6. Change the Bid size at the price MD Price Level 1 changed from 5 MIO to 3 MIO
7. Add New Best Bid at the price MD Price Level 1 in the amount of 7
8. Delete Bid MD Price Levels 6
9. Delete 2 Offer MD Price Levels 4 & 5
10. Add new Offer of the size of 5 MIO at the price 1.0987 to the MD Price Level 4

## 11. Add new Offer of the size of 1 MIO at the price 1.0990 to the MD Price Level 5

The Incremental Refresh message will be broadcasted as following:

Rep. Gr   Tag		Tag Name	VALUE	Description
→	279	MDUpdateAction	1	<b>1 = Change.</b> Type of Market Data update action
→	269	MDEntryType	0	<b>0 = BID.</b> Type of Market Data entry
→	48	SecurityID	ID	Unique instrument ID
→	271	MDEntrySize	3	Quantity represented by the Market Data Entry.
→	270	MDEntryPx	1.0981	Price of the Market Data Entry.
→	279	MDUpdateAction	0	<b>0 = New.</b> Type of Market Data update action
→	269	MDEntryType	0	<b>0 = BID.</b> Type of Market Data entry
→	48	SecurityID	ID	Unique instrument ID
→	271	MDEntrySize	7	Quantity represented by the Market Data Entry.
→	270	MDEntryPx	1.0982	Price of the Market Data Entry.
→	279	MDUpdateAction	2	<b>2 = Delete.</b> Type of Market Data update action
→	269	MDEntryType	0	<b>0 = BID.</b> Type of Market Data entry
→	48	SecurityID	ID	Unique instrument ID
→	271	MDEntrySize	6	Quantity represented by the Market Data Entry.
→	270	MDEntryPx	1.0975	Price of the Market Data Entry.
→	279	MDUpdateAction	2	<b>2 = Delete.</b> Type of Market Data update action
→	269	MDEntryType	1	<b>1 = OFFER.</b> Type of Market Data entry
→	48	SecurityID	ID	Unique instrument ID
→	271	MDEntrySize	10	Quantity represented by the Market Data Entry.
→	270	MDEntryPx	1.0987	Price of the Market Data Entry.
→	279	MDUpdateAction	2	<b>2 = Delete.</b> Type of Market Data update action
→	269	MDEntryType	1	<b>1 = OFFER.</b> Type of Market Data entry
→	48	SecurityID	ID	Unique instrument ID
→	271	MDEntrySize	20	Quantity represented by the Market Data Entry.
→	270	MDEntryPx	1.0989	Price of the Market Data Entry.
→	279	MDUpdateAction	0	<b>0 = New.</b> Type of Market Data update action
→	269	MDEntryType	1	<b>1 = OFFER.</b> Type of Market Data entry
→	48	SecurityID	ID	Unique instrument ID
→	271	MDEntrySize	5	Quantity represented by the Market Data Entry.
→	270	MDEntryPx	1.0987	Price of the Market Data Entry.
→	279	MDUpdateAction	0	<b>0 = New.</b> Type of Market Data update action
→	269	MDEntryType	1	<b>1 = OFFER.</b> Type of Market Data entry
→	48	SecurityID	ID	Unique instrument ID

Rep. Gr   Tag		Tag Name	VALUE	Description
→	271	MDEntrySize	5	Quantity represented by the Market Data Entry.
→	270	MDEntryPx	1.0990	Price of the Market Data Entry.

After applying this incremental refresh the Level 2 Book looks as following:

LEVEL 2 AGGREGATED DEPTH BOOK			
BID		ASK	
Quantity	Price	Price	Quantity
7	1.0982	1.0983	5
3	1.0981	1.0984	14
12	1.098	1.0986	12
8	1.0979	1.0987	5
14	1.0978	1.099	1

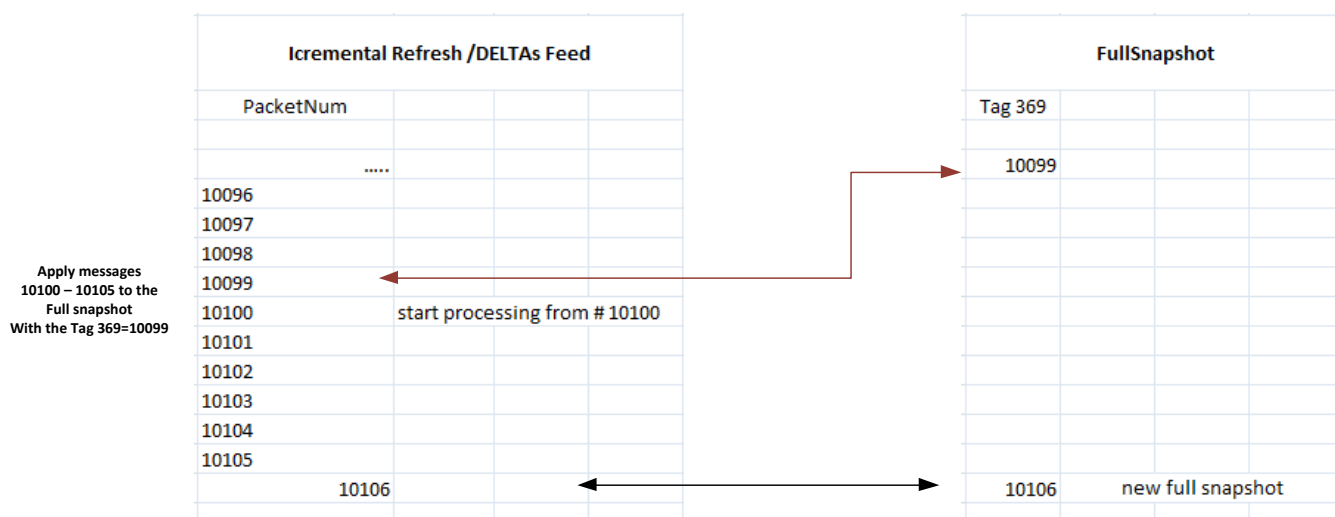
The sequence of the book update should be in the order presented in the message.

## Chapter 8 Managing ‘Late Joiners’

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

- Download the configuration files and schema files from the Thomson Reuters ftp site.
- Listen to the Instrument Definition Feed.
- Listen to the Statistics Feed
- Listen to the Incremental feed for incremental (deltas) market data. Begin queuing messages.
- Wait for the latest Snapshot (35=W) message; when the latest snapshots is received, verify that the value for tag 369-LastMsgSeqNumProcessed in the Snapshot (35=W) message matches the latest packet sequence number from the Incremental feed and, additionally, the value for tag 83-RptSeq in the Snapshot (35=W) message matches tag 83-RptSeq in the Incremental feed.
- If there is a mismatch between tag 369-LastMsgSeqNumProcessed in the Snapshot (35=W) message and the latest packet sequence number from the Incremental feed (and/or tag 83-RptSeq in the Incremental feed), then wait for the next Snapshot (35=W) message while still queuing incremental Refresh messages. After the respective match confirmed, discarded all the queued incremental Refresh packets.

The diagram below illustrates the flow:



## Chapter 9 Pre-Opening Startup

For a startup prior to the weekly market open, all market data including Book updates, statistics, instrument definitions, trades will be disseminated through all the respective feeds defined above.

Follow the process below to ensure that all necessary market data is received:

1. Download the configuration files and schema files from the Thomson Reuters ftp site.
2. Listen to the Instrument Definition Feed.
3. Listen for the first Incremental Refresh (35='X')
4. Wait and listen for the first Snapshot (35='W') message
5. Verify that the Book was correctly created as per description above.
6. Start to apply Incremental Refresh messages to the Book.
7. Listen to the Statistics Feed /Statistics Feed is out of the scope in the Phase 1 Release/



## Chapter 10 Data Dictionary

The table below provides a Data Dictionary for all the fields used in this document.

- The 'offset' attribute is not provided for known fields as the offset is different on the different messages.

Tag #	Tag Name	Definition
48	SecurityID	<field name="SecurityIDSource" id="22" type="SecurityIDSource" description="Identifies class or source of tag 48-SecurityID value" semanticType="char"/>
60	TransactTime	<field name="TransactTime" id="60" type="uint60" description="Start of event processing time." semanticType="UTCTimestamp"/>
64	SettlDate	<field name="LastUpdateTime" id="64" type="uint64" description="Timestamp of when the instrument was last added, modified or deleted" semanticType="UTCTimestamp"/>
75	TradeDate	<field name="TradeDate" id="75" type="LocalMktDate" description="Trade Session Date" semanticType="LocalMktDate"/>
83	RptSeq	<field name="RptSeq" id="83" type="uint32" description="Sequence number per instrument update" semanticType="int"/>
167	SecurityType	<field name="SecurityType" id="167" type="SecurityType" description="Security Type" semanticType="String"/>
264	MarketDepth	<field name="MarketDepth" id="264" type="Int8" description="Identifies the depth of book" semanticType="int"/>
268	NoMDEntries	<group name="NoMDEntries" id="268" description="Number of entries in Market Data message" blockLength="33" dimensionType="GroupSize">
269	MDEntryType	<field name="MDEntryType" id="269" type="MDEntryTypeBook" description="Market Data entry type" semanticType="char"/>
270	MDEntryPx	<field name="MDEntryPx" id="270" type="PRICE" description="Trade price" semanticType="Price"/>
271	MDEntrySize	<field name="MDEntrySize" id="271" type="Int32" description="Trade quantity" semanticType="Qty"/>
279	MDUpdateAction	<field name="MDUpdateAction" id="279" type="MDUpdateAction" description="Market Data update action" semanticType="int"/>
282	MDEntryOriginator	<field name="MDEntryOriginator" id="282" type="MDEntryOriginator" description="originator of the market data" offset="41" semanticType="String"/>
369	LastMsgSeqNumProcessed	<field name="LastMsgSeqNumProcessed" id="369" type="uint32" description="Sequence number of the last Incremental feed packet processed. This value is used to synchronize the snapshot loop with the real-time feed" semanticType="SeqNum"/>
562	MinTradeVol	<field name="MinTradeVol" id="562" type="uint32" description="The minimum trading volume for a security" offset="83" semanticType="Qty"/>
779	LastUpdateTime	<field name="LastUpdateTime" id="779" type="uint64" description="UTC Date and time of last Security Definition add, update or delete on a given Market Data channel" semanticType="UTCTimestamp"/>

Tag #	Tag Name	Definition
828	TrdType	<field name="TrdType" id="828" type="uInt8" description="Type of trade" semanticType="int"/>
943	TimeBracket	VWAP
980	Security Update Action	<field name="SecurityUpdateAction" id="980" type="SecurityUpdateAction" description="Last Security update action on Incremental feed, 'D' or 'M' is used when a mid-week deletion or modification (i.e. extension) occurs" semanticType="char"/>
1020	TradeVolume	<field name="MDEntrySize" id="1020" type="Int32" description="Cumulative traded volume" semanticType="Qty"/>
1180	ApplID	<field name="ApplID" id="1180" type="Int16" description="Indicates the channel ID as defined in the XML configuration file sinceVersion="3" semanticType="int"/>
30375	Currency1 [BASECUR]	<field name="Currency1" id="30375" type="Currency" description="Identifies currency used for price" semanticType="Currency"/>
30376	Currency2	<field name="Currency2" id="30376" type="Currency" description="Id entifies currency used for price" semanticType="Currency"/>
30378	BasisPoint	<type name="BasisPoint" description="BasisPoint" primitiveType="uint8"/>
30379	RatePrecision [PIPSIZE]	<type name="RatePrecision" description="RatePrecision" primitiveType="uint8"/>
30380	RateTerm	VWAP: HHMMSS TimeOfDay
30381	Currency1AmtDecimals	<type name="Currency1AmtDecimals" description="Currency1AmtDecimals" primitiveType="uint8"/>
30382	Currency2AmtDecimals	<type name="Currency2AmtDecimals" description="Currency2AmtDecimals" primitiveType="uint8"/>
30430	TradesFeedConflation Interval	msec
30431	SecurityDefinition Conflation Interval	msec
30432	AggressorSide	<field name="AggressorSide" id="30432" type="LegSide" description="Buyside or Sellside" semanticType="UInt8"/>
30433	RGTSMDPS	
30434	LEFT_DPS	Price Mantissa, exponent = -7 (price is multiplied by 10million)
30435	RIGHT_DPS	Price Mantissa, exponent = -7 (price is multiplied by 10million)
30436	CLS	Y or N
30437	SnapshotConflationInterval	msec
30438	IncRefreshConflation Interval	msec
30439	DepthOfBook	<field name="DepthOfBook" id="30439" type="uInt8" description="Type of Book for which the feed is intended." semanticType="int"/>

© 2016 Thomson Reuters. All rights reserved.  
Republication or redistribution of Thomson Reuters content, including by framing or similar means, is prohibited without the prior written consent of Thomson Reuters. 'Thomson Reuters' and the Thomson Reuters logo are registered trademarks and trademarks of Thomson Reuters and its affiliated companies.

**For more information**

Send us a sales enquiry at  
[financial.thomsonreuters.com/sales](http://financial.thomsonreuters.com/sales)

Read more about our products at  
[financial.thomsonreuters.com](http://financial.thomsonreuters.com)

Find out how to contact your local office  
[financial.thomsonreuters.com/locations](http://financial.thomsonreuters.com/locations)

