National Commodity and Derivatives Exchange

Market Data Feed

UDP FIX/FAST Specification

Version	4.01
Release Date	06 Oct 2017

1 DOCUMENT CONTROL

1.1 Table of Contents

1.2 Revision History

Date	Version	Sections	Description
21 Apr 2015	1.00		Initial Draft
05 Nov 2015	2.00		Add Datatype and Length
20 May 2016	3.00		A new enum introduced to MDStatType (30101) to disseminate instrument level statistics and description provided. Trading halt reason as a result of a price band breach included. Length of Headline (148) and Text (58) changed. NCDEXExpiryDate (22010) introduced to disseminate NCDEX Expiry Date via Security Definition message.
02 Jun 2017	4.00	3.7.1 6.5.1.1 6.5.3.1 6.5.4.1 7.2 7.3	Addition of option greeks and IV Added new tags related to option instruments Addition of option greeks and IV Updated CFI code table Updated security type
06 Oct 2017	4.01	7.1 6.4.1	Clarification added with regards to segment Clarification added with regards to security definition request send for specific segment

1.3 Exclusions

The given functionality is currently not supported by NCDEX

Sr. NO	Functionality	
1	Order Types	Market
		Market If Touched
		Trailing Stop
		Trailing Stop Limit
		Reserve
		Minimum Fill
		Named
		At the Open (OPG)
		Fill or Kill(FOK)
		At the Close (ATC)

		Good For Auction(GFA)
		Good Till Time (GTT)
		Closing Price Cross
2	Give-up from CTCL Front end	
3	Negotiated Trades	

1.4 References

FAST 1.1 Session Control Protocol Specification
FIX 5.0 (Service Pack 2) Specification

1.5 Definitions, Acronyms and Abbreviations

ASR	Active Spread Range. Defines the range within which an order can be accepted to a Calendar Spread Order Book, where executions are possible.
Client	A recipient connected to the Snapshot or Replay channel of the market data feed.
FAST	Version 1.1of the Session Control Protocol of the FIX Adapted for Streaming specification.
FIX	Version 5.0 (Service Pack 2) of the Financial Information Exchange Protocol.
Orders	Executable interest in the order book.
Recipient	A subscriber to the market data feed.
Server	The market data interface of NCDEX.
Sub Book	Each instrument is traded across multiple separate and distinct sub books (e.g. regular, off book.). Messages transmitted on the feed include an indication of the instrument and sub book to which it relates.
VWAP	Volume weighted average price.

2 OVERVIEW

The market data feed is a stream of FAST encoded FIX messages which provides the following real-time information for each instrument traded on NCDEX:

(i) Price depth information for the order book. The feed provides information on the aggregated displayed quantity and the number of displayed orders for each of the top *<five>* price points.

(ii) Order depth information for the order book. The feed provides information on the price and displayed quantity of each order in the top *<five>* price points.

(iii) Details (e.g. price, volume, time, etc.) of on-book trades.

(iv) Statistics (e.g. high/low, volume, VWAP, etc.).

(v) Trading status.

Each instrument is traded on a series of separate and independent sub books (e.g. regular) The above information is disseminated per instrument and sub book combination. An update transmitted on the feed includes an indication of the instrument and sub book to which it relates.

In addition, the feed includes market and sector statistics as well as the value of each index computed by NCDEX. It also provides participants with the active instrument list of and disseminates market announcements.

The feed is a multicast service based on the technology and industry standards UDP, IPv4, FAST and FIX. The application messages are defined using the FIX 5.0 (Service Pack 2) standard and comply with the best practices outlined by the FIX Market Data Working Group. Please refer to Section 6.1 for the instances where the server varies from the FIX protocol. The data feed is transmitted in the FAST encoding method to minimize bandwidth and reduce latency and conforms to Level 1 of the FAST 1.1 specification.

The exchange requires that the vendor/exchange's member undergo a conformance test upon completing development of the interface. The vendor/exchange's member must contact the exchange to schedule an appropriate period for testing.

The vendor/exchange's member may contact the CTCL division of the exchange to seek clarification

CTCL Team (Technology Department)
National Commodity & Derivatives Exchange Limited,
Akruti Corporate Park, 1st Floor, Near G.E.Garden, L.B.S. Marg,
Kanjurmarg (West), Mumbai - 400 078
Phone (022) 66406717/66406609-12
Fax (022) 66473201/66406899
Email: CTCL@ncdex.com

3 SERVICE DESCRIPTION

3.1 System Architecture

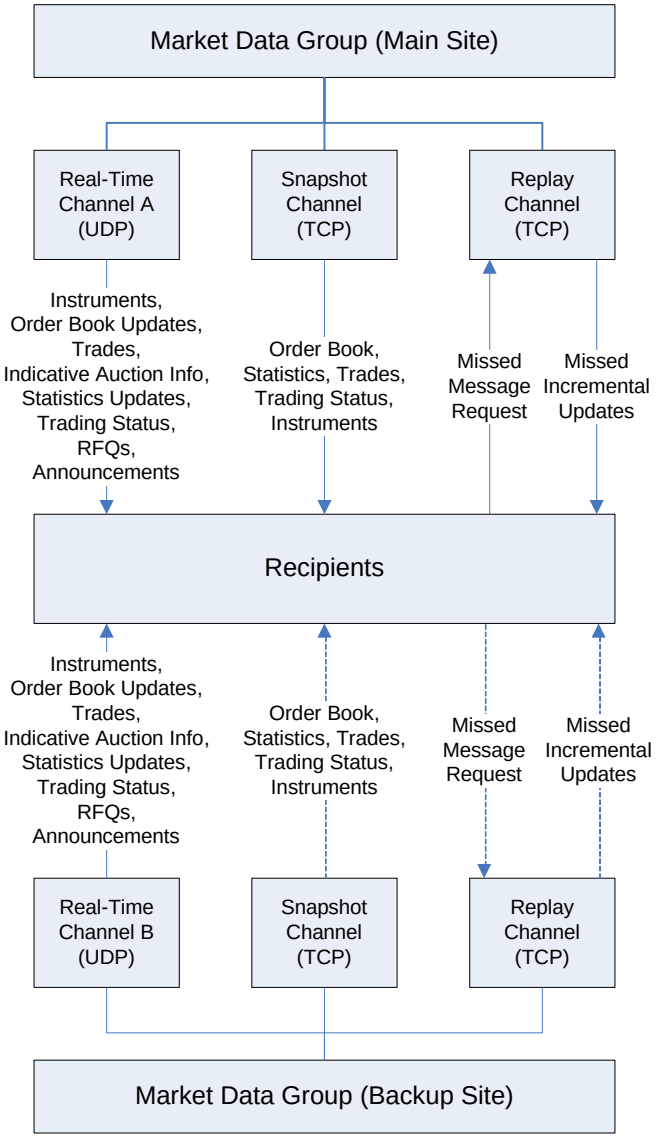
The market data feed is load balanced by market data group.

While each group will contain multiple instruments, each instrument is assigned to just one market data group. Although the group an instrument is assigned to may change from day to day, it will not change within a day. Market data for all sub books (e.g. regular) for a particular instrument are transmitted from the same market data group.

Each market data group includes a multicast Real-Time channel for the dissemination of market data.

Two TCP recovery channels are available per market data group; Snapshot and Replay.

While a recipient may connect to the Replay channel to recover from a small data loss, it should use the Snapshot channel after a large data loss (i.e. late joiner or major outage).



3.1.1 Real-Time Channel

The Real-Time Channel is the primary means of disseminating market data. Real-time updates to instruments and all market data supported by the feed are available on this multicast channel.

The list of instruments[1] in the market data group is broadcasted at the start of the trading day via the Security Definition message. The details of instruments created during trading hours will also be disseminated in via this message. Real-time updates of the trading status of instruments will be disseminated via the Security Status message.

Real-time updates to order books and statistics are published along with the details of each trade via the Market Data Incremental Refresh message. While each Market Data Incremental Refresh includes a channel specific message sequence number in the field ApplSeqNum (1181), each market data entry in the message includes an instrument specific sequence number in the field RptSeq (83). The channel and instrument level sequence numbers are to reset to 1 at the start of each day.

This channel will also disseminate announcements via News messages.

The server will use the Heartbeat message to exercise the communication line during periods of inactivity. A Heartbeat will be sent every <2> seconds when the Real-Time channel is inactive. Each Heartbeat message includes the next expected application sequence number (ApplSeqNum (1181) of the next application message) in the field AppNewSeqNum (1399).

[1] Based on the Exchange configuration the list may contain all instruments or only the active instruments in the market data group

Recipients have access to two identically sequenced Real-Time feeds; one from the main site (Feed A) and one from the backup site (Feed B). It is recommended that recipients process both feeds and arbitrate between them to minimise the probability of a data loss.

3.1.2 Snapshot Channel

The TCP Snapshot channel permits recipients to request a snapshot of the order book and statistics for any instrument in the market data group as well as its current trading status. In addition, it enables recipients to request the retransmission of the trades published during the last *<10>* minutes on the Real-Time channel. It also enables recipients to download the list of instruments in the market data group. This channel may be used by recipients to recover from a large-scale data loss.

All messages sent by the server are transfer encoded in terms of the FAST protocol. While all application messages sent by the server (e.g. Market Data Snapshot (Full Refresh)) are field encoded, the administrative messages it sends (e.g. Logon, Heartbeat, etc.) are not. All messages (i.e. both administrative and application) initiated by the client should be transfer encoded but not field encoded.

While a Snapshot channel is available from the backup site, it will only be activated in the unlikely event of an outage at the main site.

3.1.3 Replay Channel

The TCP Replay channel permits recipients to request the retransmission of a limited number of messages already published on the Real-Time channel. This channel may be used by recipients to recover from a small data loss.

The Replay channel supports the retransmission of the last *<10,000>* messages published on the Real-Time channel. The channel does not support the retransmission of messages published on the Snapshot channel or from previous trading days.

All messages sent by the server are transfer encoded in terms of the FAST protocol. While all application messages sent by the server (e.g. Market Data Incremental Refresh, Security Definition, etc.) are field encoded, the administrative messages it sends (e.g. Logon, Heartbeat, etc.) are not. All messages (i.e. both administrative and application) initiated by the client should be transfer encoded but not field encoded.

While a Replay channel is available from the backup site, it will only be activated in the unlikely event of an outage at the main site.

3.2 Message Overview

The market data feed utilises the FIX application messages described below to disseminate instruments, market data and market announcements.

Message	Description	Usage (By Channel)		
		Real-Time	Snapshot	Replay
Security Definition	Used to disseminate details (e.g. status, ISIN, underlying, issuer, order books, etc.) on all instruments. Each message will only contain the details of one instrument.	√	√	√
Security Status	Used to communicate the trading status (e.g. Regular Trading, Halt, etc.) of instruments.	√	×	√
Market Data Incremental Refresh	Used to provide: (i) An update to the order book. (ii) Information on an executed on-book trade. (iii) Book-level statistics (e.g. low, high, volume, VWAP, etc.). (iv) Market, sector and Instrument Type statistics (e.g. volume, turnover, etc.) (v) Index values A single message may contain multiple market data entries which may cover one or more instruments.	√	√	√
Market Data Snapshot	Used to disseminate a full snapshot of the order book and statistics for an instrument.	√	√	√
News	Used to publish market announcements.	√	×	√

3.3 Overview of a Trading Day

3.3.1 Trading on the Order Book

The regular day for on-book trading will consist of four scheduled sessions: Pre-Trading, Regular Trading, Closing Price Publication, and Post-Close. The start time for each of these sessions may vary from one set of instruments to another. A Security Status message will be published on the Real-Time channel to indicate when a particular session has commenced for a sub book of an instrument .An overview of the trading day is given below.

Session	Description
Market Open	The market data feed begins. Recipients should aim to join the feed at this time. <*Two*> minutes after the market opens, a Security Definition message will be broadcast for each instrument on the Real-Time channel. The closing value of each index at the end of the previous trading day will also be published at this time. No new orders, cancel requests and modification requests will be accepted during this time. A Security Status message will not be broadcast at Market Open.
Pre-Trading	Participants may not submit or modify orders during this session. However they are allowed to cancel carry forward orders. Market operations may cancel or modify orders. The details of GTD/GTC orders carried over from the previous trading day will be disseminated at the start of this session. Order book updates will be disseminated.

Session	Description
Regular Trading	Continuous trading begins for the regular order book. Order book updates, trades, statistics and index updates will be disseminated.
Closing Price Publication	The official closing price of each instrument will be calculated. Participants may not submit, cancel or modify orders during this session.
Post-Close	While order cancel requests will be accepted, no new orders and modification requests may be submitted during this session. Order book updates will be disseminated. At the end of this session a Security Status message will be broadcast for each tradable instrument with End of Post-Close (103) as the SecurityTradingStatus (326).
Market Close	The end of the trading day. A Security Status message will be broadcast for each sub book of each active instrument with Market Close (18) as the SecurityTradingStatus (326).
End of Day	The feed will stop broadcasting messages at this time.

3.3.2 Trade Reporting

At the start of the reporting period a Security Status message, with a SecurityTradingStatus (326) of Regular Trading (17) and an MDSubBookType (1173) of Block Trade (4), will be broadcast for each instrument for which trade reporting is permitted.

At the start of the reporting period a Security Status message, with a SecurityTradingStatus (326) of Regular Trading (17) and an MDSubBookType (1173) of Block Trade (4), will be broadcast for each instrument for which trade reporting is permitted. Similarly, a Security Status message, with a SecurityTradingStatus (326) of Market Close (18) and an MDSubBookType (1173) of Block Trade (4), will be broadcast for each such instrument at the end of the reporting period.

3.3.3 Reporting Leg Trades

To denote the leg trades, the Market Data Incremental Refresh message broadcasts an MDUpdateAction (279) of New (0) and MDEntryType (269) of Trade (2). The trade price, quantity and time is specified in the fields MDEntryPx (270), MDEntrySize (271) and MDEntryTime (273) respectively.

3.3.4 Trading Halt

NCDEX may institute two types of trading halts for an instrument: a temporary halt that will not carry over to the next day or a longer term halt that spans across trading days (e.g. regulatory halt).

3.3.4.1 Temporary Halt

A Security Status message, with a SecurityTradingStatus (326) of Halt (2), will be published if an instrument is temporarily halted, due to the trigger of a price band or imposed manually by Market Operations. The reason for the halt and whether it applies to a particular sub book (e.g. regular) or to trade reporting will be specified in the HaltReason (327) and MDSubBookType (1173) fields respectively. If trading is later resumed a Security Status message, with the appropriate status (e.g. Regular Trading (17)) and MDSubBookType (1173), will be published. A temporary halt will not be carried forward to the next trading day.

3.3.4.2 Longer Term Halt

A Security Definition message, with a SecurityStatus (965) of Halted (8), will be published if an instrument is halted across multiple days. A Security Status, with a SecurityTrading Status (326) of Halt (2), will also be published for each sub book associated with the instrument (i.e. regular, EFP trade).

If, at the start of a trading day, an instrument is still in a halted state it will be included in the Security Definition messages published by the server. A Security Status, with a SecurityTradingStatus (326) of Halt (2), will also be published for each associated sub book at the start of the first scheduled session (i.e. Pre-Trading).

A Security Definition message, with a SecurityStatus (965) of Active (1), will be published if the halt is lifted during trading hours. Separate Security Status messages will also be published for each order book that moves in to a trading session.

3.3.5 Pause

A Security Status message, with a SecurityTradingStatus (326) of Pause (111), will be published if on-book trading for an instrument is paused during the day. A Security Status message, with the appropriate status (e.g. Regular Trading) will be published if trading is later resumed. A pause will not be carried forward to the next trading day.

3.3.6 Instrument Suspension

An instrument may be suspended during or outside trading hours. The suspension may be lifted later in the day or it may be carried forward to subsequent trading days. A Security Definition message, with a SecurityStatus (965) of Suspended (9), will be published if an instrument is suspended during trading hours. A suspension applies to both on-book trading and trade reporting.

If, at the start of a trading day, an instrument is still in a suspended state it will be included in the Security Definition messages published by the server but not the Security Status messages. A Security Definition message, with a SecurityStatus (965) of Active (1), will be published if the suspension is lifted during trading hours. Separate Security Status messages will also be published for each order book that moves in to a trading session.

3.3.7 Intra-Day Trading Session Updates

3.3.7.1 Adjustment by Market Operations

NCDEX may extend or shorten a particular trading session. In such a case, a Security Status message will be broadcast with the value Extended by Market Operations (101) or Shortened by Market Operations (102) in the field SecurityTradingEvent (1174). The message will indicate whether the change applies to a particular sub book (e.g. regular) or to trade reporting. The new time at which the session will end will be specified in the Text (58) field.

3.3.8 New Instruments

New instruments may be created during the trading day. In such a case, to notify the recipients of the details of the new instrument (symbol, order books, segment, underlying, expiration date, etc.) the server will publish a separate Security Definition message for the instrument creation[2] and for each order book associated with the instrument upon sequential creation of order books. Separate Security Status messages will also be published for each order book that moves to a trading session upon creation.

[2] The first Security Definition message sent for the instrument creation will have no order books attached.

3.4 Order Book Management (Price Depth)

The market data feed provides recipients with a view of the regular order book, where all orders are aggregated at each price level, for *<five>* price levels. The feed provides the aggregate displayed quantity and the number of represented orders at each price level. This information is broadcast as incremental updates on the Real-Time channel.

3.4.1 Incremental Refresh

A Market Data Incremental Refresh message will be published to update an order book. A message may contain multiple market data entries each of which could add, change or delete one side of one price level. Each entry includes the fields MDSubBookType (1173) and MDPriceLevel (1023) to indicate the sub book and price level being updated. A single message may include entries for multiple instruments and sub books. MDBookType(1021) of Market Data Incremental Refresh message will be PriceDepth(2) to indicate that the message is published via price depth service.

3.4.1.1 Adding a Price Level

When a new price level is created in an order book a Market Data Incremental Refresh message will be broadcast with an MDUpdateAction (279) of New (0). The message will include the price, aggregate displayed quantity and number of orders for the price level.

The field MDPriceLevel (1023) will indicate the display position of the price level being added. Price levels are numbered from most to least competitive and start with “1”.All rows in the order book below the new price level should be pushed down. The recipient’s application should automatically re-number the price levels below the newly added price level. If there were already *<five>* price levels, recipients should also delete the new *<sixth>*price level from their applications.

3.4.1.2 Changing a Price Level

When an existing price level is changed a Market Data Incremental Refresh message will be broadcast with an MDUpdateAction (279) of Change (1). The field MDPriceLevel (1023) will indicate which price level is being updated. The message will include the price, quantity and number of orders for the updated price level.

3.4.1.3 Deleting a Price Level

When an existing price level is removed a Market Data Incremental Refresh message will be broadcast with an MDUpdateAction (279) of Delete (2). The field MDPriceLevel (1023) will indicate which price level is being removed.

All rows in the order book below the deleted price level should be pushed up. The recipient’s application should automatically re-number the price levels that were below the deleted price level. The server will separately publish an update to add a new price level at the bottom of the order book.

3.4.2 Snapshot

Market Data Snapshot (Full Refresh) messages will be used to disseminate the full picture of each order book. The snapshot will include the price, aggregate display quantity, number of orders and display position for each price level.

A snapshot will be published even if there are no orders in the order book. In such a case, the Market Data Snapshot (Full Refresh) message will include an MDEntryType (269) of Empty Order Book (J).

An order book snapshot may span across multiple Market Data Snapshot (Full Refresh) messages. In such cases, the LastFragment (893) field will be “N” for all but the final message

for the instrument. The final message will include a LastFragment (893) of “Y”. MDBookType(1021) of Market Data Snapshot (Full Refresh) message will be PriceDepth(2) to indicate that the message is published via price depth service.

The vast majority of snapshot messages are disseminated via the Snapshot channel. However in certain scenarios (e.g. recovery after a server failure), they may also be included in the Real-Time channel. Snapshots on the Real-Time channel should be processed by recipients.

3.5 Order Book Management (Order Depth)[3]

The market data feed provides recipients with a view of first *<five>*orders of the order book. The feed provides the price and displayed quantity of each order. This information is broadcast as *<<incremental updates> or <a snapshot every one second if there is an update>>* on the Real-Time channel.

3.5.1 Incremental Refresh

A Market Data Incremental Refresh message will be published to update an order book. A message may contain multiple market data entries each of which could add, change or delete an order. A single message may include entries for multiple instruments and sub books.

Each entry includes an indication of the sub book and an identifier of the order in the fields MDSubBookType (1173) and MDEntryID (278) respectively .As the order identifier is the OrderID (37) assigned to it by the matching system, recipients will be able to identify their own orders while ensuring anonymity. Order IDs, and therefore MDEntryIDs, are unique across all instruments and across trading days.

MDBookType (1021) of Market Data Incremental Refresh message will be OrderDepth (2) to indicate that the message is published via order depth service.

3.5.1.1 Adding an Order

When a new order is added to the order book a Market Data Incremental Refresh message will be broadcast with an MDUpdateAction (279) of New (0). The message will include the price and displayed quantity of the order.

While the field MDPriceLevel (1023) will indicate the display position of the price level the order is added to, the field MDEntryPositionNo (290) will indicate its position within the price level. Price levels are numbered from most to least competitive and start with “1”. Similarly, position numbers within a price level are numbered from most to least competitive and start with “1” for each price level.

If the order is added at a new price level, all rows in the order book below the new price level should be pushed down. The recipient’s application should automatically re-number the price levels below the newly added price level. If there were already *<five>* price levels, recipients should also delete the new *<sixth>* price level from their applications.

If, in terms of its MDEntryPositionNo (290), an order is added ahead of some existing orders at the price level, these orders should be pushed down. The recipient’s application should automatically re-number the position numbers of orders at the price level if they are below the newly added order.

If MDPriceLevel (1023) is set to “1” the message will update the top of the order book. The recipient’s application should ensure that there are no prices higher than this price level. Similarly, if MDEntryPositionNo (290) is set to “1” the message will update the top of the price level. The recipient’s application should ensure that there are no orders higher than this order at the price level.

3.5.1.2 Updating an Order

A Market Data Incremental Refresh will be broadcast with an MDUpdateAction (279) of Change (1) if an existing order is updated without a loss of priority (e.g. a reduction in order quantity). The field MDEntryID (278) will indicate which order is being updated. The message will include the order’s price and updated display quantity. The relevant MDPriceLevel (1023) and MDEntryPositionNo (290) will also be provided.

If the update of an existing order results in a loss of priority (e.g. an increase in order quantity, a change in price, etc.), the update will be communicated via two actions; a delete (as outlined below) followed by an addition.

[3]Delete this section if the market data feed is configured for price depth.

3.5.1.3 Deleting an Order

If an existing order is removed a Market Data Incremental Refresh will be broadcast with an MDUpdateAction (279) of Delete (2). The field MDEntryID (278) will identify the order being removed.

Orders at the price level of the deleted order should be pushed up if they were below it. The recipient's application should automatically re-number the position numbers of such orders.

If the order was the only order at the price level, all rows in the order book below the deleted price level should be pushed up. The recipient's application should automatically re-number the price levels that were below the deleted price level. The server will separately publish updates to add orders at the new price level at the bottom of the order book.

3.5.2 Snapshot

Market Data Snapshot (Full Refresh) messages will be used to disseminate the full picture of each order book. The snapshot will include the price and displayed quantity of each order along with the display position of each price level and order.

A snapshot will be published even if there are no orders in the order book. In such a case, the Market Data Snapshot (Full Refresh) message will include an MDEntryType (269) of Empty Order Book (J).

An order book snapshot may span across multiple Market Data Snapshot (Full Refresh) messages. In such cases, the LastFragment (893) field will be "N" for all but the final message for the instrument. The final message will include a LastFragment (893) of "Y"

While the vast majority of snapshot messages are disseminated via the Snapshot channel, in certain scenarios (e.g. recovery after a server failure) they may also be included in the Real-Time channel. Snapshots on the Real-Time channel should be processed by recipients.

MDBookType (1021) of Market Data Snapshot (Full Refresh) message will be OrderDepth (2) to indicate that the message is published via price depth service.

3.6 Time and Sales

The market data feed provides recipients with the price, volume, time and sub book (e.g. regular) for each executed trade. Details of trade cancellations and corrections will also be broadcast.

When a trade is executed a Market Data Incremental Refresh message will be broadcast with an MDUpdateAction (279) of New (0) and MDEntryType (269) of Trade (2). The trade price, quantity, time and sub book will be specified in the fields MDEntryPx (270), MDEntrySize (271), MDEntryTime (273) and MDSubBookType (1173) respectively.

Each trade will include a unique Trade ID in the field MDEntryID (278) which will be referenced if the trade is cancelled or corrected. As this identifier is the TradeID (1003) assigned to the trade by the post trade system, recipients will be able to identify their own trades. Trade IDs are unique across all instruments and across trading days.

3.6.1 Pre-Negotiated Trades

The Market Data Incremental Refresh message will include an MDSubBookType (1173) of EFP Trade (4) in the case of a pre-negotiated trade.

3.6.2 Trade Cancellation and Corrections

If a trade is cancelled a Market Data Incremental Refresh message will be broadcast with an MDUpdateAction (279) of Delete (2) and an MDEntryType (269) of Trade (2). The message will contain the Trade ID of the cancelled trade in the field MDEntryID (278).

If a trade is corrected a Market Data Incremental Refresh message will be broadcast with an MDUpdateAction (279) of Change (1) and an MDEntryType (269) of Trade (2). The message

will contain the Trade ID of the corrected trade, the corrected price and the corrected quantity in the fields MDEntryID (278), MDEntryPx (270) and MDEntrySize (271) respectively.

A trade cancellation is final (i.e. once a trade is cancelled it will not be reinstated). A corrected trade may subsequently be corrected again or cancelled.

3.7 Statistics

3.7.1 Book-Level Statistics

The market data feed provides recipients with the statistics listed below for the sub books of each instrument. Each statistic applies only to the current day and to the specified sub book.

Statistics	Regular	Block Trade	Relevant Fields
Opening price	√	×	MDEntryType (269) = 4 MDEntryPx (270)
Closing price	√	×	MDEntryType (269) = 5 MDEntryPx (270)
Highest traded price	√	√	MDEntryType (269) = 7 MDEntryPx (270)
Lowest traded price	√	√	MDEntryType (269) = 8 MDEntryPx (270)
VWAP	√	√	MDEntryType (269) = 9 MDEntryPx (270)
Volume	√	√	MDEntryType (269) = B MDEntrySize (271)
Turnover	√	√	MDEntryType (269) = x MDEntryPx (270)
Number of trades	√	√	MDEntryType (269) = y MDEntrySize (271)
Highest bid price	√	×	MDEntryType (269) = N MDEntryPx (270)
Lowest offer price	√	×	MDEntryType (269) = O MDEntryPx (270)
Previous Close	√	×	MDEntryType (269) = f MDEntryPx (270)
Buy Order VWAP	√	×	MDEntryType (269) = s MDEntryPx (270)
Sell Order VWAP	√	×	MDEntryType (269) = t MDEntryPx (270)
Buy Order Qty	√	×	MDEntryType (269) = u MDEntrySize (271)
Sell Order Qty	√	×	MDEntryType (269) = v MDEntrySize (271)
Open Interest	√	×	MDEntryType (269) = C MDEntrySize (271)
Change in Open Interest	√	×	MDEntryType (269) = (p) MDEntrySize (271)
Opening Open Interest	√	×	MDEntryType (269) = (n) MDEntrySize (271)
Closing Open Interest	√	×	MDEntryType (269) = (o) MDEntrySize (271)
Day High Open Interest	√	×	MDEntryType (269) = (q) MDEntrySize (271)

Day Low Open Interest	√	×	MDEntryType (269) = (r) MDEntrySize (271)
TheoreticalPrice	√	×	MDEntryType (269) = (h) MDEntryPx (270)
Delta	√	×	MDEntryType (269) = (i) MDEntryPx (270)
Gamma	√	×	MDEntryType (269) = (j) MDEntryPx (270)
Vega	√	×	MDEntryType (269) = (k) MDEntryPx (270)
Theta	√	×	MDEntryType (269) = (l) MDEntryPx (270)
Rho	√	×	MDEntryType (269) = (m) MDEntryPx (270)
ImpliedVolatility	√	×	MDEntryType (269) = (w) MDEntryPx (270)

3.7.2 Market, Sector and Instrument Type Statistics

The feed also provides recipients with the market, sector and Instrument type statistics listed below. The MDStatType (31001) field will be used to indicate whether the statistic being published relates to a market or sector or an instrument type (it will not be present in the case of a statistic for a sub-book of an instrument). The market or sector or instrument type will be identified via the Symbol (55) field.

Statistics	Market	Sector	Instrument Type	Relevant Fields
Volume for the day	√	√	√	MDEntryType (269) = B MDEntrySize (271)
Turnover for the day	√	√	√	MDEntryType (269) = x MDEntryPx (270)
Number of trades for the day	√	√	√	MDEntryType (269) = y MDEntrySize (271)

3.7.3 Instrument level Statistics

Book level statistics will be (Number of trades, Volume and Turnover) amalgamated at instrument level to receive updates at the same. The MDStatType (31001) field in the Market Data Incremental Refresh message will be used to indicate whether the statistic being published relates to an instrument level and the instrument will be identified via the Symbol (55) field.

Statistics	Instrument Level	Relevant Fields
Volume for the day	√	MDEntryType (269) = B MDEntrySize (271)
Turnover for the day	√	MDEntryType (269) = x MDEntryPx (270)
Number of trades for the day	√	MDEntryType (269) = y MDEntrySize (271)

3.7.4 Upper Active Spread Range and Lower Active Spread Range

The ASR provides the market an indication of what price range an order can be entered into the calendar spread instrument, where executions are possible.

An MDEntryType (269) of UASR (K) and LASR (L) will indicate the upper and lower ASR values respectively. This information will be disseminated via Market Data Incremental Refresh message and Market Data Snapshot (Full Refresh) message on FAST real time channel.

3.7.5 Spot Price of an Underlying

The spot price of underlying instruments will be broadcast via the Market Data Incremental Refresh message with an MDEntryType (269) of Spot Price (D)

The date and time of discovery of the spot price is communicated via the MDEntryDate (272) and MDEntryTime (273) fields respectively This information will be disseminated via Market Data Incremental Refresh message and Market Data Snapshot (Full Refresh) message on FAST real time channel.

3.7.6 Publication of Open Interest Statistics

The following open interest related statistics generated by StarGate for future and underlying instruments will be sent out via the FAST real time channel as current statistics for the day.

- Open Interest
- Change in Open Interest
- Closing Open Interest
- Opening Open Interest
- Day High Open Interest
- Day Low Open Interest

3.7.7 Mode of Dissemination

3.7.7.1 Incremental Update

Incremental updates of statistics will be disseminated via the Market Data Incremental Refresh message. The MDEntryType (269) and MDSubBookType (1173) of each entry in such a message will indicate the statistic that is being updated and the sub book to which it applies. The MDUpdate Action (279) of all entries will be New (0).

In the event a statistic (e.g. closing price) needs to be corrected, a Market Data Incremental Refresh message will be transmitted with the corrected value. The entry will not include an MDEntryPx (270) or MDEntrySize (271) in the unlikely event a previously published statistic is withdrawn (e.g. closing price).

3.7.7.2 Snapshot

Snapshots of statistics will be disseminated via the Market Data Snapshot (Full Refresh) message. The MDEntryType (269) and MDSubBookType (1173) of each entry in such a message will indicate the statistic that is being updated and the sub book to which it applies.

A snapshot will be published even if there are no statistics for an instrument. In such a case, the Market Data Snapshot (Full Refresh) message will include an MDEntryType (269) of No Statistics (z).

While the vast majority of snapshot messages are disseminated via the Snapshot channel, in certain scenarios (e.g. recovery after a server failure), they may also be included in the Real-Time channel. Snapshots on the Real-Time channel should be processed by recipients.

3.8 Indices

The market data feed provides recipients with the value of each index

When the value of an index is updated a Market Data Incremental Refresh message will be broadcast with an MDUpdateAction (279) of New (0) and an MDEntryType (269) of Index Value (3). The unique identifier and updated value of the index will be specified in the fields Symbol (55) and MDEntryPx (270) respectively.

The value of each index will be updated to reflect the closing price of each constituent instrument once these prices are published. Each such index update will include a TradeCondition (277) of Closing Index (AJ)[4].

The closing value of each index from the previous trading day will be published soon after the market opens. Each such update will include a TradeCondition (277) of Previous Closing Index (ZZ).

3.9 Announcements

Market announcements are included in the market data feed. In addition to the actual text of the announcement and an associated headline or subject, recipients are provided with its urgency, the time it was generated and the list of instruments and underlying instruments, if any, to which it relates.

Announcements are disseminated via the News message.

[4] Delete this paragraph if the feature to update indices based on closing prices is disabled.

4 CONNECTIVITY

4.1 Transmission Standards

4.1.1 Multicast Channels

The Real-Time channel utilises UDP over IP version 4 (IPv4) and Ethernet standards. UDP header information will be as defined in the IETF RFC 791 (IPv4) and RFC 768 (UDP) transmission protocol standards. One or more FAST encoded FIX messages may be included in a single UDP packet.

4.1.2 Point-to-Point Channels

The Snapshot and Replay channels utilise TCP over IP version 4 (IPv4) and Ethernet standards. TCP header information will be as defined in the IETF RFC 793 standard and IPv4 will be as defined in the RFC 791 standard.

4.2 Application IDs

4.2.1 Clients

The CompID of each client wishing to connect to the Snapshot and Replay channels must be registered with NCDEX before communications can begin. A CompID may, at any particular time, only be logged into one Snapshot channel and one Replay channel across all market data groups.

4.2.1.1 Passwords

Each new CompID will be assigned a password on registration. Clients are strongly encouraged to change the password to one of their choosing via the Logon message. The acceptance of a login request indicates that the new password has been accepted. The new password will, if accepted, be effective for subsequent logins.

In terms of the password policy of NCDEX, the password of each CompID should be changed at least every *<30>* days. If not, the password will expire and the client will be unable to login to the server. In such a case, the client should contact NCDEX to have its password reset. The SessionStatus (1409) of the server's Logon message will be Password Due to Expire (2) for the last *<5>* days of a password's validity period.

4.3 Production IP Addresses and Ports

The market data feed is load balanced by market data group. While each group will contain multiple instruments, each instrument is assigned to just one market data group. Although the group an instrument is assigned to may change from day to day, it will not change within a day. The Security Definition messages available on the Real-Time channel of the various market data groups may be utilized by recipients to identify the instruments assigned to each group.

The production IP addresses and ports of the Real-Time, Snapshot and Replay channels for each market data group will be shared before FIX communications can begin.

5 RECOVERY

5.1 Recipient Failures

It is recommended that recipients process both Real-Time feeds (i.e. Feed A and Feed B) to minimise the probability of a data loss.

A message loss can be detected using the ApplSeqNum (1181) included in each message on the Real-Time channel. If a gap in sequence numbers is detected, the recipient should assume that some or all of the order books and statistics maintained on its systems are incorrect and initiate one of the recovery processes outlined below.

Each entry of a Market Data Incremental Refresh message disseminated on the Real-Time channel also includes an instrument specific sequence number in the field RptSeq (83). Recipients may use this instrument level sequencing to determine the instruments for which a recovery process should be initiated.

5.1.1 Snapshot Channel

The TCP Snapshot channel should be used by recipients to recover from a large-scale data loss (i.e. late joiner or major outage).

The channel permits recipients to request a snapshot of the order books and book-level statistics for the instruments in the market data group. In addition, it enables recipients to request the retransmission of trades published during the last *<10>* minutes on the Real-Time channel. It also enables recipients to download the list of instruments in the market data group. Snapshots of index values, market statistics and sector statistics are not available.

Each CompID may login to the Snapshot channel of a particular market data group up to *<5>* times each day. The total number of Security Definition Request and Market Data Request messages that a client may submit on the Snapshot channel of a particular market data group is also limited to*<100>*each day. Recipients may request NCDEX to reset its login and request counts. This feature is intended to help manage an emergency situation and should not be relied upon as a normal practice.

If a client submits multiple requests on the Snapshot channel, they will be processed serially (i.e. one at a time). Active requests of multiple clients will be served on a round robin basis. A client may not have more than *<10>* concurrent unprocessed requests at any point.

All messages sent by the server are transfer encoded in terms of the FAST protocol. While all application messages sent by the server (e.g. Market Data Snapshot (Full Refresh), Security Definition, etc.) are field encoded, the administrative messages it sends (e.g. Logon, Heartbeat, etc.) are not. All messages (i.e. both administrative and application) initiated by the client should be transfer encoded but not field encoded.

5.1.1.1 Establishing a Connection

The client should use the relevant IP address and port (as outlined in Section 0) to establish a TCP/IP session with the Snapshot channel. The client should initiate a connection by sending the Logon message. The client should identify itself by specifying its CompID in the Username (553) field.

The server will validate the CompID, password and IP address of the client.Once the client is authenticated, the server will respond with a Logon message. The SessionStatus (1409) of this message will be Session Active (0).

The client must wait for the server's Logon before sending additional messages. Messages received from the client before the exchange of Logon messages will be ignored.

If a logon attempt fails because of an invalid CompID, password or IP address, the server will break the TCP/IP connection with the client without sending a Logout message. If a logon attempt fails because of an expired password, a locked CompID or if logins are not currently permitted, the server will send a Logout and then break the TCP/IP connection with the client.

The Snapshot channel supports only <100> concurrent logins across all clients. In addition, each CompID may login to the Snapshot channel of a particular market data group up to<5> times each day. If either of these limits is reached, the server will reject a new logon attempt with a Logout and then break the TCP/IP connection with the client. The SessionStatus (1409) of such a Logout message will be Logons Not Allowed (7).

If a Logon is not received within <5>[5] seconds of the establishment of a TCP/IP connection, the server will break the TCP/IP connection with the client without sending a Logout message.

If a Security Definition Request or Market Data Request is not received within <5> seconds of a successful logon, the server will send a Logout message and then break the TCP/IP connection with the client.

A second attempt to log in to the same Market Data Snapshot Channel or a Market Data Snapshot Channel of a different Market Data Group, by an already logged in client will be rejected immediately by breaking the connection. No message is sent to the client in this case, as the client is not authenticated. The original session is not affected by this disconnection.

5.1.1.2 Heartbeats

The server will not send heartbeats on the Snapshot channel during periods of inactivity.

5.1.1.3 Requesting Instruments

Once connected to the Snapshot channel, clients may use the Security Definition Request message to request the details of all instruments in the market data group or those in the group from a particular segment. The Security Definition Request should include a unique SecurityReqID (320) and a SecurityRequestType (321) (i.e. all instruments or all instruments for a segment).

If the request is successful, the server will disseminate the details of each instrument via the Security Definition message. Each such message will include a SecurityResponseType (323) of List of Securities Returned (4) and the SecurityReqID (320) of the request. The last message sent in response to the request will include a LastRptRequested (912) of Last Message (Y). The client will not receive the details of any subsequent master file updates.

An invalid Security Definition Request will generally be rejected via a Security Definition message with a SecurityResponseType (323) of Cannot Match Selection Criteria (6). However, a Business Message Reject will be used if a conditionally required field is missing or a request limit (for the day or for outstanding requests) breached.

5.1.1.4 Requesting Snapshots: Order Book and Statistics

Once connected to the Snapshot channel, clients may use the Market Data Request message to request a snapshot of the current order books or statistics for one or more instruments in the market data group. Clients may also request snapshots for the instruments in the market data group that are assigned to one or more segments. The Market Data Request should include an MDReqID (262) and at least one Symbol (55) or ProductComplex (1227). However, the messages sent back to honour a Market Data Request will not be filtered by the order book specified in MDSubBookType (1173) for a given ProductComplex (1227).

A request for an order book snapshot should include an MDEntryType (269) of Bid (0) and/or Offer (1). An order book snapshot will always contain the details both sides the order book even if the Market Data Request contained just an MDEntryType (269) of Bid (0) or Offer (1). Clients are unable to request just one side (i.e. bid or offer) of an order book.

A request for a Top of Book snapshot should include MDBookType (1021) of Top of Book(1). If MDBookType (1021) is not specified, the server will publish snapshots of MBP and MBO order books, excluding Top of Book information.

Similarly, a request for a statistics snapshot should include one or more of the relevant entries for MDEntryType (269) (e.g. High Price (7), Low Price (8), Volume (B), etc.). A statistics snapshot will contain all available statistics even if the Market Data Request contained just

[5]Insert delay limit here and press Ctrl A and then F9 to update all other references to it.

some of the statistics-related entries for MDEntryType (269). Clients are unable to request for a snapshot of selected statistics (e.g. high price and low price only).

The MDSubBookType (1173) field of the Market Data Request may be used if the client wishes to recover the order book or statistics for a specific sub book (e.g. regular). If anMDSubBookType (1173) is not included in a request, the server will interpret it as a request for all of the sub books that apply to the instrument.

A client may request both an order book snapshot and a statistics snapshot in a single Market Data Request. A single message may also be used to request snapshots for multiple instruments and/or multiple segments.

If the Market Data Request is successful, the server will disseminate a snapshot of the current order book or statistics for each sub book via a Market Data Snapshot (Full Refresh) message. Each such message will include the MDReqID (262) of the applicable Market Data Request. The order book and statistics snapshot for a particular sub book of an instrument will be disseminated via two separate Market Data Snapshot (Full Refresh) messages.

Each Market Data Snapshot (Full Refresh) will reflect the order book or statistics for the sub book of an instrument as per the last update on the Real-Time channel. The ApplSeqNum (1181) and RptSeq (83) of the final Market Data Incremental Refresh(That is the final order book update to any instrument.) from the Real-Time channel will be included in the LastMsgSeqNumProcessed (369) and RptSeq (83) fields of each Market Data Snapshot (Full Refresh) respectively.

An order book or statistics snapshot will be published even if there are no orders or statistics for a sub book of an instrument. In such a case, the Market Data Snapshot (Full Refresh) message will include an MDEntryType (269) of Empty Order Book (J) or No Statistics (z).

Each Market Data Snapshot (Full Refresh) message whether it is used to provide a snapshot of the order book or statistics, will include the current trading status (e.g. regular trading) of the sub book in the field MDSecurityTradingStatus (1682).

The last message sent in response to the request will include a LastRptRequested (912) of Last Message (Y). The client will not receive any subsequent order book or statistics updates for the applicable instruments.

An invalid Market Data Request will generally be rejected via a Market Data Request Reject message. However, a Business Message Reject will be used if a conditionally required field is missing or a request limit (for the day or for outstanding requests) breached.

5.1.1.5 Requesting Missed Trades

Once connected to the Snapshot channel, clients may use the Market Data Request message to request missed trades for one or more instruments in the market data group. Clients may also request missed trades for the instruments in the market data group that are assigned to one or more segments. The Market Data Request should include an MDReqID (262) and at least one Symbol (55) or ProductComplex (1227). However, the missed trades re-transmitted to honour a Market Data Request will not be filtered by the order book specified in MDSubBookType (1173) for a given ProductComplex (1227).

Clients are unable to request for trades relating to a specific sub book (e.g. regular). A request for missed trades will result in the re-transmission of trades for all sub books that apply to the instrument.

A Market Data Request for missed trades should include an MDEntryType (269) of Trade (2). The request should also include the sending time of the last trade on the Real-Time channel processed by the client in the MDEntryTime (273) field.

A client may request missed trades and order book and/or statistics snapshots in a single Market Data Request. A single message may also be used to request missed trades for multiple instruments and/or multiple segments.

If the Market Data Request is successful, the server will retransmit all the trades for the specified instruments and/or segments after the specified time. Each missed trade will be published via a Market Data Incremental Refresh message. Each such message will include

the MDReqID (262) of the applicable Market Data Request. The corresponding message and instrument sequence numbers from the Real-Time channel will be included in the ApplSeqNum (1181) and RptSeq (83) fields of each retransmitted trade. The last message sent in response to the request will include a LastRptRequested (912) of Last Message (Y).

If the request includes an MDEntryTime (273) that is prior to that of the oldest trade in the server's cache, all eligible trades in the cache will be retransmitted. Clients are unable to recover trades not in the server's cache.

An invalid Market Data Request will generally be rejected via a Market Data Request Reject message. If no trades have been missed, the Market Data Request Reject will include an MDReqRejReason (281) of "Z" and the value "102" (i.e. requested market data unavailable) in the Text (58) field.

A Market Data Request will be rejected via a Business Message Reject if a conditionally required field is missing or a request limit is breached.

5.1.1.6 Requesting Missed Announcements

Once connected to the Snapshot channel, clients may use the Market Data Request message to request for missed News messages. The MDEntryType (269) of Market Data RequestMarket Data Request should be set to News (w).

A client may request missed Announcements trades, order book and/or statistics snapshots in a single Market Data Request.

Upon submitting a request for News messages to the server the client will receive all the News messages for the specified instruments and/or segments after the time specified in MDEntryTime (273). The last message sent in response to the request will include a LastRptRequested (912) of Last Message (Y).ApplSeqNum (1181) of News messages retransmitted to honour a Market Data Request will be set to zero.

If no trades have been missed, the server will transmit a Market Data Request Reject message. It will include an MDReqRejReason (281) of "Z" and the value "102" (i.e. requested market data unavailable) in the Text (58) field.

The NoRelatedSymbols(146) block will not be required if the MDEntryType(269) is News(w). If specified, the values will be ignored by the server.

5.1.1.7 Cancelling a Request

A client may cancel an outstanding instrument or market data request via the Security Definition Request or Market Data Request respectively. Such a message should include a SubscriptionRequestType (263) of Unsubscribe (2) and the SecurityReqID (320) or MDReqID (262) of the request to be cancelled. While the server will not confirm a successful cancellation, it will transmit a Business Message Reject if the request is rejected.

5.1.1.8 Termination of the Connection

If the client does not send a Logout and terminate the connection within *<5>*seconds of the transmission of the last application message, the server will send a Logout message and then break the TCP/IP connection with the client.

The server will terminate the TCP/IP connection (a Logout will not be sent) if the number of messages that are buffered for a client exceeds *<1,000>*.

5.1.2 Replay Channel

The TCP Replay channel should be used by recipients to recover from a small-scale data loss. It permits recipients to request the retransmission of a limited number of messages already published on the Real-Time channel. The channel supports the retransmission of the last *<10,000>* messages published on the Real-Time channel.

Each CompID may login to the Replay channel of a particular market data group up to <10> times each day. The total number of Application Message Requests that a client may send on the Replay channel of a particular market data group is also limited to<10>each day. Recipients may request NCDEX to reset its login and request counts. This feature is intended to help manage an emergency situation and should not be relied upon as a normal practice.

If a client submits multiple requests on the Replay channel, they will be processed serially (i.e. one at a time). Active requests of multiple clients will be served on a round robin basis. A client may not have more than <1> concurrent unprocessed requests at any point.

All messages sent by the server are transfer encoded in terms of the FAST protocol. While all application messages sent by the server (e.g. Market Data Incremental Refresh, Security Definition, etc.) are field encoded, the administrative messages it sends (e.g. Logon, Heartbeat, etc.) are not. All messages (i.e. both administrative and application) initiated by the client should be transfer encoded but not field encoded.

5.1.2.1 Establishing a Connection

The client should use the relevant IP address and port (as outlined in Section 0) to establish a TCP/IP session with the Replay channel. The client should initiate a session by sending the Logon message. The client should identify itself by specifying its CompID in the Username (553) field.

The server will validate the CompID, password and IP address of the client. Once the client is authenticated, the server will respond with a Logon message. The SessionStatus (1409) of this message will be Session Active (0).

The client must wait for the server's Logon before sending additional messages. Messages received from the client before the exchange of Logon messages will be ignored.

If a logon attempt fails because of an invalid CompID, password or IP address, the server will break the TCP/IP connection with the client without sending a Logout. If a logon attempt fails because of an expired password, a locked CompID or if logins are not currently permitted, the server will send a Logout and then break the TCP/IP connection with the client.

The Replay channel supports only <100> concurrent logins across all clients. In addition, each CompID may login to the Replay channel of a particular market data group up to<10>times each day. If either of these limits is reached, the server will reject a new logon attempt with a Logout and then break the TCP/IP connection with the client. The SessionStatus (1409) of such a Logout message will be Logons Not Allowed (7).

If a Logon is not received within <5> seconds of the establishment of a TCP/IP connection or an Application Message Request is not received within<5>seconds of a successful logon, the server will send a Logout and then break the TCP/IP connection with the client.

A second attempt to log in to the same Market Data Replay Channel or a Market Data Replay Channel of a different Market Data Group, by an already logged in client will be rejected immediately by breaking the connection. No message is sent to the client in this case, as the client is not authenticated. The original session is not affected by this disconnection.

5.1.2.2 Heartbeats

The server will not send heartbeats on the Replay channel during periods of inactivity.

5.1.2.3 Requesting Missed Messages

The client is expected to transmit an Application Message Request within<5>seconds of establishing the connection.

The message should include the identifier of the Real-Time channel to which the retransmission request applies along with the list of messages to be resent. The ApplBegSeqNum (1182) and ApplEndSeqNum (1183) fields should be used to specify the ApplSeqNum (1181) of the first and last message in the range to be resent.

The Application Message Request can be used in three modes:

(i) To request a single message. The ApplBegSeqNum (1182) and ApplEndSeqNum (1183) should both be the message sequence number of the missed message.

(ii) To request a specific range of messages. The ApplBegSeqNum (1182) should be the message sequence number of the first message of the range and the ApplEndSeqNum (1183) should be that of the last message of the range.

(iii) To request all messages after a particular message. The ApplBegSeqNum (1182) should be the message sequence number immediately after that of the last processed message and the ApplEndSeqNum (1183) should be zero (0).

The retransmission request will be serviced from the server's cache of the last *<10,000>* messages published on the Real-Time channel. If the retransmission request includes one or more messages that are not in the server's cache, the entire request will be rejected and no messages will be retransmitted.

5.1.2.4 Response to a Retransmission Request

The server will respond to the Application Message Request with an Application Message Request Ack to indicate whether the retransmission request is successful or not. If the request is unsuccessful, the reason will be specified in the field ApplResponseType (1348).

An Application Message Request will be rejected via a Business Message Reject if a conditionally required field is missing or a request limit (for the day or for outstanding requests) is breached.

In the case of a successful retransmission request, the server will retransmit the requested messages immediately after theApplication Message Request Ack. The message sequence number and, where relevant, the instrument sequence number from the Real-Time channel will be included in the ApplSeqNum (1181) and RptSeq (83) fields of each retransmitted message. Once the last of these messages is sent, the server will indicate that the retransmission is complete via an Application Message Report.

5.1.2.5 Cancelling of a Retransmission Request

A client may cancel an outstanding retransmission request via an Application Message Request. Such a message should include an ApplReqType (1347) of Cancel Retransmission (5) and the ApplReqID (1346) of the request to be cancelled. The server will transmit an Application Message Request Ack or a Business Message Reject to confirm or reject the cancellation request respectively.

5.1.2.6 Termination of the Connection

If the client does not send a Logout and terminate the connection within*<5>*seconds of the transmission of the last missed message, the server will send a Logout message and then break the TCP/IP connection with the client.

The server will terminate the TCP/IP connection (a Logout will not be sent) if the number of messages that are buffered for a client exceeds *<1,000>*.

5.2 Failures at NCDEX

5.2.1 Snapshots on the Real-Time Channel

In the unlikely event of an outage of a system at NCDEX, recipients may be required to refresh their order book and statistics displays for one or more instruments.

In such a scenario the server will, on the Real-Time channel, broadcast two Market Data Snapshot (Full Refresh) messages for each sub book of each affected instrument: one with an MDEntryType (269) of Empty Order Book (J) and the other with an MDEntryType (269) of No Statistics (z). In such an event, recipients must discard the contents of their order book and statistics displays. Each Market Data Snapshot (Full Refresh) message will include the current status of on-book trading for the instrument in the field MDSecurityTradingStatus (1682).

The server will then transmit a series of Market Data Incremental Refresh messages to disseminate the current order book and statistics for the sub books of each affected instrument.

5.2.2 Resetting Sequence Numbers

If the market data feed is, due to the unlikely event of an outage, restarted during a trading day, the message sequence number and instrument level sequence number of the Real-Time channel will be reset to 1.

In such a case, messages sent on the Real-Time channel prior to the resetting of sequence numbers will not be available for retransmission on the Replay channel. The trades resent on the Snapshot channel, if any, will not contain an ApplSeqNum (1181) or RptSeq (83) if there were originally disseminated prior to the resetting of sequence numbers.

6 MESSAGE FORMATS AND TEMPLATES

This section provides details on the header, three administrative messages and six teen application messages utilized by the market data feed.

All messages sent by the server are transfer encoded in terms of the FAST protocol. While all application messages sent by the server (e.g. Market Data Incremental Refresh, Security Definition, etc.) are field encoded, the administrative messages it sends (e.g. Logon, Heartbeat, etc.) are not. All messages (i.e. both administrative and application) initiated by the client should be transfer encoded but not field encoded.

The FIX format of each is described along with the applicable FAST template.

In certain scenarios, a single event in the matching system will result in the publication of multiple messages on the market data feed. NCDEX may change the order in which these messages are published at any time without prior notice. Recipients should ensure that their applications are developed to process each of the messages covered in this section irrespective of the order in which they are transmitted.

NCDEX reserves the right to add new message types without prior notice. Recipients should develop their decoders to ignore unknown message types.

6.1 Variations from the FIX Protocol

The market data feed conforms to the FIX protocol except as follows:

(i) The Market Data Snapshot (Full Refresh) and Market Data Incremental Refresh messages include the custom field OpenCloseIndicator (30002). The data type of this field is String.

(ii) The Market Data Incremental Refresh message includes the custom field MDStatType (31001). The data type of this field is Int (i.e. integer).

(iii) The Market Data Request message includes the field MDEntryTime (273).

(iv) The Market Data Request message does not include the field MarketDepth (264).

(v) The MDSubBookType (1173) field is included in the Market Data Request and Security Status messages.

(vi) The MDEntryType (269) field of the Market Data Snapshot (Full Refresh) message includes the custom value No Statistics (z).

(vii) The TradeCondition (277) field of the Market Data Incremental Refresh messages includes the custom value Previous Closing Index (ZZ).

(viii) The MDEntryType (269) field of the Market Data Snapshot (Full Refresh) and Market Data Incremental Refresh messages includes the values Market Bid (b) and Market Offer (c) which were introduced in Extension Pack 106.

(ix) The MDEntryType (269) field of the Market Data Snapshot (Full Refresh) and Market Data Incremental Refresh messages includes the custom values Previous Close (f), AON Statistics (o), Buy Order VWAP (s), Sell Order VWAP (t), Buy Order Qty (u) and Sell Order Qty (v).

(x) The Market Data Snapshot (Full Refresh) message includes the fields MDSecurityTradingStatus (1682) and MDHaltReason (1684) which were introduced in Extension Pack 106.

(xi) TrdType (828) and TrdSubType (829) is included in the Market Data Incremental Refresh message.

(xii) The LastRptRequested (912) field is included in the Security Definition, News, Market Data Snapshot (Full Refresh) and Market Data Incremental Refresh messages.

(xiii) The SecurityStatus (965) field of the Security Definition message includes the custom values Halted (8) and Suspended (9).

(xiv) The SecurityTradingEvent (1174) field of the Security Status message includes the custom values Extended by Market Operations (101) and Shortened by Market Operations (102).

(xv) The MDReqRejReason (281) field of the Market Data Request Reject message includes the custom value Other (Z).

(xvi) The HaltReason (327) field of the Security Status message contains custom values specific to NCDEX.

(xvii) NoMDEntries(268) repeating group, which includes MDSubBookType(1173) field, is included in the Security Definition message.

(xviii) The MDEntryType (269) field of Market Data Incremental Refresh and Market Data Snapshot (Full Refresh) messages includes the custom fields Change in Open Interest (C), Opening Open Interest (n), Closing Open Interest (o), Change in Open Interest (p), Day High Open Interest (q) and Day Low Open Interest (r). Respective values will be communicated via MDEntrySize (271) field.

(xix) The MDEntryType (269) field of Market Data Incremental Refresh and Market Data Snapshot (Full Refresh) messages includes the custom fields Upper Active Spread Range (K) and Lower Active Spread Range (L).

(xx) The SendingTime (52) field in the Header message disseminates the 'Update Time' of an instrument.

(xxi) The Security Definition message fields ContractMultiplier (231), UnitofMeasure (996). PriceUnitOfMeasure (1191), SecurityDesc (107), MinTradeVol (562), MaxTradeVol (1140), RoundLot (561), DeliveryForm (668), IssueDate (225) and TradingReferencePrice (1150) are used to communicate Contract Multiplier, Quantity Unit, Price Unit, Security Description, Minimum Size, Max Qty, Lot Size, Delivery Type, First Trading Date and the Reference Price of the Instrument.

(xxii) TickIncrement (1208) field within the NoTickRules (1205) repeating block is used to disseminate the Tick Size of the Instrument.

(xxiii) The InstrAttribType (871) field within the NoInstrAttrib (870) repeating block of the Security Definition message includes custom fields PriceBandPercentage (120), Delivery Lot (121), Delivery Unit (122) and Margin Percent (123).

(xxiv) Market Data Incremental Refresh and Market Data Snapshot (Full Refresh) messages include custom fields; the MDEntryType (269) field is used to communicate Price Band (g), while the new fields LowLimitPrice (1148), HighLimitPrice (1149), PriceLimitEvent (22006), SecurityTradingStatus (326), PriceBandLimits (22008), PriceLimitUpdate (22007) and PriceLimitType (1306) will be introduced.

6.2 Header

6.2.1 FIX Message

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req</th><th>Length</th><th>Description</th></tr>
<tr><td>35</td><td>MsgType</td><td>Y</td><td>2</td><td>Message type.
Value | Meaning
0 | Heartbeat
5 | Logout
A | Logon
B | News
V | Market Data Request
W | Market Data Snapshot (Full Refresh)
X | Market Data Incremental Refresh
Y | Market Data Request Reject
BW | Application Message Request
BX | Application Message Request Ack
BY | Application Message Report
c | Security Definition Request
d | Security Definition
f | Security Status
j | Business Message Reject</td></tr>
<tr><td>52</td><td>SendingTime</td><td>Y</td><td>21</td><td>Time the message is transmitted specified in UTC and in the YYYYMMDD-HH:MM:SS.sss format.</td></tr>
</table>

6.3 Administrative Messages

6.3.1 Logon

6.3.1.1 FIX Message

Tag	Field Name	Req	Length	Description
Header				
1180	ApplID	N	N/A	Identifier of the server sending the message. Required if the message is generated by the server.
108	HeartBtInt	Y	UI32	Indicates the heartbeat interval in seconds.
553	Username	N	12	CompID of the client. Required if the message is generated by the client.
554	Password	N	30	Password assigned to the CompID. Required if the message is generated by the client.
925	NewPassword	N	30	New password for the CompID.
1409	SessionStatus	N	UI32	Status of session. Required if message is generated by server. **Value** – **Meaning** 0 – Session Active 2 – Password Due to Expire

6.3.2 Logout

6.3.2.1 FIX Message

Tag	Field Name	Req	Length	Description
Header				
1180	ApplID	N	N/A	Identifier of the server sending the message. Required if the message is generated by the server.
1409	SessionStatus	N	UI32	Status of the session. Required if the message is generated by the server. **Value** – **Meaning** 4 – Session Logout Complete 6 – Account Locked 7 – Logons Not Allowed 8 – Password Expired 100 – Other
58	Text	N	61	Reason for the logout.

6.3.3 Heartbeat

6.3.3.1 FIX Message

Tag	Field Name	Req	Length	Description
Header				
1180	ApplID	Y	N/A	Identifier of the server sending the message.
1399	AppNewSeqNum	Y	I32	The next expected application sequence number (ApplSeqNum (1181) of the next application message)

6.4 Application Messages (Client-Initiated)

6.4.1 Security Definition Request

6.4.1.1 FIX Message

Tag	Field Name	Req	Length	Description
Header				
320	SecurityReqID	Y	N/A	Identifier of the request.
263	Subscription RequestType	Y	I32	Whether the request is being cancelled. **Value** **Meaning** 0 Snapshot 2 Unsubscribe
321	SecurityRequest Type	N	I32	Type of request. Required unless Subscription RequestType (263) is Unsubscribe (2). **Value** **Meaning** 8 All Securities 9 All Securities for a Segment
1300	MarketSegmentID	N	31	Indicates the segment. Please refer to Section 7.1 for the valid segments. Required if SecurityRequest Type (321) is All Securities for a Segment (9). Please refer revised segment list in section 7.1. As per the list, security definition can be requested for the desired instrument types.

6.4.2 Market Data Request

6.4.2.1 FIX Message

Tag	Field Name		Req	Length	Description
Header					
262	MDReqID		Y	N/A	Identifier of the market data request.
263	Subscription RequestType		Y	I32	Type of request. **Value** **Meaning** 0 Snapshot 2 Unsubscribe
267	NoMDEntryTypes		N	I32	Number of market data types requested. Required if Subscription RequestType (263) is Snapshot (0).
➡	269	MDEntry Type	N	1	Indicates the type of market data requested. Required if NoMDEntryTypes (267) is specified. **Value** **Meaning** 0 Bid 1 Offer 2 Trade 4 Opening Price 5 Closing Price 7 High Price 8 Low Price 9 VWAP B Volume D Spot Price (Spot price of the Commodity) K UASR (Upper Active Spread Range) L LASR (Lower Active Spread Range) N Highest Bid Price O Lowest Offer Price b Market Bid c Market Offer g Price Band f Previous Close x Turnover y Number of Trades w News
➡	273	MDEntry Time	N	21	Sending time of the last processed trade. The time should be specified in UTC and in the HH:MM:SS.sss format. Required if MDEntryType (269) is Trade (2).

146	NoRelatedSym		N	I32	Number of instruments or segments market data is requested for. Required if SubscriptionRequest Type (263) is Snapshot (0).
➡	55	Symbol	N	30	Unique identifier of the instrument. Required if ProductComplex (1227) is not specified.
➡	1227	Product Complex	N	31	Indicates the segment. Please refer to Section 7.1 for the valid segments. Required if Symbol (55) is not specified.
➡	1173	MDSubBook Type	N	1	Type of market Order book to which the request relates. Absence of this field should be interpreted as for all order books.[6] **Value** **Meaning** 1 Regular 4 EFP Trade

[6] However, the messages sent back to honour a Market Data Request will not be filtered by the order book specified in MDSubBookType (1173) for a given ProductComplex (1227).

6.4.3 Application Message Request

6.4.3.1 FIX Message

Tag	Field Name		Req	Length	Description
Header					
1346	ApplReqID		Y	N/A	Identifier of the request.
1347	ApplReqType		Y	I32	Type of request. **Value** **Meaning** 0 Retransmission of Messages 5 Cancel Retransmission
1351	NoApplIDs		N	I32	If specified, the value in this field should always be "1". Required if ApplReqType (1347) is Retransmission of Messages (0).
➡	1355	RefApplID	N	N/A	ApplID of the Real-Time channel for which the retransmission is requested. Please refer to Section**Error! Reference source not found.**for the list of valid ApplIDs. Required if NoApplIDs (1351) is specified.
➡	1182	ApplBeg SeqNum	N	I32	ApplSeqNum (1181) of the first message in the range to be resent from the Real-Time channel. Required if NoApplIDs (1351) is specified.
➡	1183	ApplEnd SeqNum	N	I32	ApplSeqNum (1181) of the last message in the range to be resent from the Real-Time channel. Required if NoApplIDs (1351) is specified.

6.5 Application Messages (Server-Initiated)

6.5.1 Security Definition

6.5.1.1 FIX Message

Tag	Field Name	Req	Length	Description
Header				
1180	ApplID	Y	N/A	Identifier of the server sending the message.
1181	ApplSeqNum	N	I32	Sequence number of the message on the Real-Time channel. Required if the message is disseminated via the Real-Time or Replay channel.
320	SecurityReqID	N	N/A	Identifier of the Security Definition Request this message relates to. Required if the message is disseminated via the Snapshot channel.
323	SecurityResponseType	N	I32	Type of response. Required if the message is disseminated via the Snapshot channel. **Value** **Meaning** 4 List of Securities Returned 6 Cannot Match Selection Criteria
912	LastRptRequested	N	1	Indicates the last message sent in response to a request. **Value** **Meaning** Y Last Message
55	Symbol	N	30	Unique identifier of the instrument.
22009	InternalRefFlag	N	15	Communicates the Token ID of the instrument
965	SecurityStatus	N	1	Status of the instrument. **Value** **Meaning** 1 Active 2 Inactive 8 Halted 9 Suspended
454	NoSecurityAltID	N	I32	If present, value in this field will always be “1”.
➡	455 SecurityAltID	N	11	Identification number for the security.
➡	456 SecurityAltIDSource	N	1	Type of security identification number used. Required if SecurityAltID (455) is specified. **Value** **Meaning** 4 ISIN
461	CFICode	N	1	Indicates the instrument type. Please refer to Section 7.2 for the valid CFI codes.
167	SecurityType	N	1	Indicates the instrument type. Please refer to Section 7.3 for the valid security types.

762	SecuritySubType		N	1	Indicates the type of multi-legged instrument. Please refer to Section 7.4 for the valid security sub types. Required for multi-legged instruments.
541	MaturityDate		N	9	Date an instrument expires. The date will be the local date (i.e. not in UTC) and will be specified in YYYYMMDD format.
711	NoUnderlyings		N	I32	If specified, the value in this field will always be “1”.
➡	311	Underlying Symbol	N	30	Unique identifier of the underlying instrument. Required if NoUnderlyings (711) is specified.
106	Issuer		N	31	Issuer of the instrument.
225	IssueDate		N	9	Date of which the instrument was issued. The date will be the local date (i.e. not in UTC) and will be specified in YYYYMMDD format.
555	NoLegs		N	I32	Number of legs for a strategy instrument.
➡	600	LegSymbol	N	31	Unique identifier of the leg instrument. Required if NoLegs (555) is specified.
➡	623	LegRatioQty	N	(20,8)	Quantity of this leg relative to that of other legs. Required if NoLegs (555) is specified.
➡	624	LegSide	N	1	Side this leg represents from the point of view of the purchaser of the strategy. Required if NoLegs (555) is specified. **Value** **Meaning** 1 Buy 2 Sell
1310	NoMarketSegments		N	I32	The value in this field will always be “1”.
➡	1300	Market SegmentID	N	31	Segment the instrument is assigned to. Please refer to Section 7.1 for the valid segments.
1151	SecurityGroup		N	31	Sector the instrument is assigned to.
870	NoInstrAttrib		N	I32	Number of instrument attributes. If present, value in this field will always be “1”.
➡	871	InstrAttribType	N	3	Codes to represent the type of instrument attribute. **Value** **Meaning** 100 Issued Quantity 120 Price Band Percentage 121 Delivery Lot 122 Delivery Unit 123 Margin Percent
➡	872	InstrAttribValue	N	10	Attribute value appropriate to the InstrAttribType (871) field.
268	NoMDEntries		N	I32	Number of market data entries within the repeating block.

➡	1173	MDSubBookType	N	1	Type of trading to which the update related. Absence of this field should be interpreted as Regular (1) **Value** **Meaning** 1 Regular 4 EFP Trade
231	ContractMultiplier		N	(15,4)	Contract Multiplier for the contract.
996	UnitOfMeasure		N	10	The unit of measure of the underlying commodity upon which the contract is based.
1191	PriceUnitOfMeasure		N	10	Used to express the UOM of the price of the contract.
107	SecurityDesc		N	31	Optional field to describe the name of the instrument.
1205	NoTickRules		N	1	If present, value in this field will always be “1”.
➡	1208	TickIncrement	N	(20,8)	Tick increment for stated price range. Specifies the valid price increments at which a security can be quoted and traded.
562	MinTradeVol		N	(17,5)	The minimum order quantity that can be submitted for an order.
1140	MaxTradeVol		N	(30,10)	The maximum order quantity that can be submitted for a security.
561	RoundLot		N	(19,7)	Trading lot size of security
668	DeliveryForm		N	10	Identifies the type of delivery.
225	IssueDate		N	9	First Trading Date of the Instrument.
1150	TradingReferencePrice		N	(15,4)	Reference Price of the Instrument.
22010	NCDEXExpiryDate		N	9	The NCDEX Expiry Date. The date will be the local date (i.e. not in UTC) and will be specified in YYYYMMDD format. Required for future and calendar spreads.
1826	EventTimePeriod		N	I32	Value specified in Max order Duration parameter
22003	MinDisclosedPercentage		N	(20,2)	The minimum percentage that has to be disclosed out of the total quantity, for an Iceberg order.
41054	NoDeliveryScheduleSettlTimes		N	1	If present, value in this field will always be “1”.
➡	41055	DeliveryScheduleSettlStart	N	9	Required if NoDeliveryScheduleSettlTimes (41054) > 0. Delivery Start Date of the Instrument
➡	41056	DeliveryScheduleSettlEnd	N	9	Required if NoDeliveryScheduleSettlTimes (41054) > 0. Delivery End Date of the Instrument

<table>
<tr><td>1198</td><td>List Method</td><td>N</td><td>4</td><td>Indicates whether the instrument is pre-listed in the system or was defined via a user request. Required for strategies, futures and options.
Value Meaning
0 Pre Listed
1 User Defined</td></tr>
<tr><td>201</td><td>PutOrCall</td><td>N</td><td>1</td><td>Required for options instruments.
Value Meaning
0 Put
1 Call</td></tr>
<tr><td>202</td><td>StrikePrice</td><td>N</td><td>(11,4)</td><td>Strike price of an options instrument. Required for options instruments.</td></tr>
<tr><td>1194</td><td>ExerciseStyle</td><td>N</td><td>1</td><td>Exercise style of an options instrument. Required for options instruments.
Value Meaning
0 European
1 American</td></tr>
</table>

6.5.2 Security Status

6.5.2.1 FIX Message

Tag	Field Name	Req	Length	Description
Header				
1180	ApplID	Y	N/A	Identifier of the server sending the message.
1181	ApplSeqNum	Y	I32	Sequence number of message on Real-Time channel.
55	Symbol	Y	30	Unique identifier of the instrument.
22009	InternalRefFlag	N	15	Communicates the Token ID of the instrument
326	SecurityTradingStatus	Y	I32	Current trading session for the instrument. **Value** **Meaning** 2 Halt 17 Regular Trading 18 Market Close 26 Post-Close 100 Pre-Trading 103 End of Post Close 111 Pause 113 Order Entry 130 Reserved for backward compatibil 125 Closing Price Publication 199 No Active Session
1174	SecurityTradingEvent	N	I32	Indicates the reason a trading session is extended or shortened. **Value** **Meaning** 1 Market Order Extension 100 Price Monitoring Extension 101 Extended by Market Operations 102 Shortened by Market Operations 103 Circuit Breaker Tripped
1173	MDSubBookType	N	I32	Sub book the update relates to. Absence of this field should be interpreted as Regular (1). **Value** **Meaning** 1 Regular 4 EFP Trade
58	Text	N	61	Time the trading session will end in the case of a session extension/shortening or if available, circuit breaker breach. The time will be specified in UTC and in the HH:MM:SS format.

327	HaltReason	N	I32	Reason for the trading halt. Please refer to Section 8 for an explanation of the reason codes. Required if SecurityTradingStatus (326) is Halt (2).

6.5.3 Market Data Snapshot (Full Refresh)

6.5.3.1 FIX Message

Tag	Field Name	Req	Length	Description
Header				
1180	ApplID	Y	N/A	Identifier of the server sending the message.
1181	ApplSeqNum	N	I32	Sequence number of message on the Real-Time channel. Required if it is disseminated via the Real-Time or Replay channel.
1021	MDBookType	N	I32	Indicates the type of service if the message is published to convey order book information. **Value** **Meaning** 2 Price Depth 3 Order Depth
893	LastFragment	N	1	Indicates the last message in a series of order book snapshot messages for the instrument. **Value** **Meaning** N Not Last Message Y Last Message
262	MDReqID	N	N/A	Identifier of the Market Data Request this message relates to. Required if the message is disseminated via the Snapshot channel.
369	LastMsgSeqNum Processed	N	I32	ApplSeqNum (1181) of the last incremental update on the Real-Time channel with which the snapshot is synchronised. Required if the message is disseminated via the Snapshot channel.
912	LastRptRequested	N	1	Indicates the last message sent in response to a request. **Value** **Meaning** Y Last Message
1173	MDSubBookType	N	I32	Sub book (Type of trading) the update relates to. Absence of this field should be interpreted as Regular (1). **Value** **Meaning** 1 Regular 4 EFP Trade
55	Symbol	Y	30	Unique identifier of the instrument.
22009	InternalRefFlag	N	15	Communicates the Token ID of the instrument

<table>
<tr><td>83</td><td>RptSeq</td><td>Y</td><td>I32</td><td>If the message is disseminated via the Snapshot channel, this field identifies the instrument specific sequence number of the last message on the Real-Time channel with which the snapshot is synchronised.
If the message is disseminated via the Real-Time or Replay channels, this field indicates the instrument specific sequence number of the Real-Time channel.</td></tr>
<tr><td>1682</td><td>MDSecurityTrading Status</td><td>N</td><td>I32</td><td>Indicates the trading status of the instrument. Required if the message is disseminated via the Snapshot channel.
Value Meaning
2 Halt
17 Regular Trading
18 Market Close
26 Post-Close
111 Pause
113 Order Entry
125 Closing Price Publication
199 No Active Session</td></tr>
<tr><td>1684</td><td>MDHaltReason</td><td>N</td><td>I32</td><td>Reason for the trading halt. Please refer to Section 8 for an explanation of the reason codes. Required if MDSecurityTradingStatus (1682) is Halt (2).</td></tr>
<tr><td>268</td><td>NoMDEntries</td><td>Y</td><td>I32</td><td>Number of market data entries in the message.</td></tr>
</table>

➡	269	MDEntryType	Y	1	Indicates the type of market data being published. This will be the first field in the repeating group.

					Value	Meaning
					0	Bid
					1	Offer
					4	Opening Price
					5	Closing Price
					7	High Price
					8	Low Price
					9	VWAP
					B	Volume
					C	Open Interest
					D	Spot Price (Spot price of the Commodity)
					J	Empty Order Book
					K	UASR (Upper Active Spread Range)
					L	LASR (Lower Active Spread Range)
					N	Highest Bid Price
					O	Lowest Offer Price
					b	Market Bid
					c	Market Offer
					f	Previous Close
					g	Price Band
					h	TheoreticalPrice
					i	Delta
					j	Gamma
					k	Vega
					l	Theta
					m	Rho
					n	Opening Open Interest
					o	Closing Open Interest
					p	Change in Open Interest
					q	Day High Open Interest
					r	Day Low Open Interest
					s	Buy Order VWAP
					t	Sell Order VWAP
					u	Buy Order Qty
					v	Sell Order Qty
					w	ImpliedVolatility

					x Turnover y Trades z No Statistics
➡	32032	PriceBandValues	N	(30,10)	Value will always be 2
➡	➡	1306 PriceLimitType	Y	1	Describes how the price limits are expressed **Value** **Meaning** 0 Price 2 Percentage
➡	➡	1148 LowLimitPrice	N	(8,0)	Value or percentage of the lower price band.
➡	➡	1149 HighLimitPrice	N	(8,0)	Value or percentage of the upper price band.
➡	278	MDEntryID	N	13	Unique identifier of each order. Required if MDEntryType (269) is Bid (0), Offer (1), Market Bid (b) or Market Offer (c) and the order book is published by order depth.
➡	270	MDEntryPx	N	(8,0)	Price of bid or offer being published. Required if MDEntryType (269) is not Volume (B), Market Bid (b), Market Offer (c), Buy Order Qty (u), Sell Order Qty (v) or Trades (y). Value may only be shown up to 8 decimals
➡	271	MDEntrySize	N	I32	Quantity of the market data entry. Required if MDEntryType (269) is not Opening Price (4), Closing Price (5), High Price (7), Low Price (8), VWAP (9), Highest Bid Price (N), Lowest Offer Price (O), Previous Close (f), Buy Order VWAP (s), Sell Order VWAP (t) or Turnover (x).
➡	272	MDEntryDate	N	9	Date on which the spot price for the Commodity was obtained. The date will be specified in DDMMYYYY format. Required if MDEntryType (269) is Spot Price (D).
➡	273	MDEntryTime	N	21	Time the spot price for the Commodity was obtained. The time will be specified in UTC and in the HH:MM:SS format. Required if MDEntryType (269) is Spot Price (D).

➡	346	NumberOfOrders		N	I32	Number of orders represented in the aggregate quantity published for a bid or offer. Required if MDEntryType (269) is Bid (0), Offer (1), Market Bid (b) or Market Offer (c) and the order book is published by price depth.
➡	290	MDEntry PositionNo		N	I32	Display position of the order in the order book within its price level. Required if MDEntryType (269) is Bid (0), Offer (1), Market Bid (b) or Market Offer (c) and the order book is published by order depth.
➡	1023	MDPriceLevel		N	I32	Display position of the price level in the order book. Required if MDEntryType (269) is Bid (0) or Offer (1).
➡	453	NoPartyIDs		N	I32	Number of party identifiers. If specified, the value in this field will always be “1”.
➡	➡	448	PartyID	N	30	Identifier of the party. Required if NoPartyIDs (453) is specified.
➡	➡	447	PartyID Source	N	1	Required if PartyID (448) is specified. **Value** **Meaning** D Proprietary/Custom Code
➡	➡	452	Party Role	N	1	Role of the specified PartyID (448). Required if PartyID (448) is specified. **Value** **Meaning** 1 Trading Firm
➡	30002	OpenClose Indicator		N	1	Method used to compute the opening or closing price. Required if MDEntryType (269) is Opening Price (4) or Closing Price (5). **Value** **Meaning** 2 Regular Trade 3 Mid-Point 6 VWAP 7 Last Regular Trade 9 Previous Close 10 Manual 19 VWAP of Last n Trades 20 Reference Price 22 Best Bid 23 Best Offer 24 Theoretical Price

6.5.4 Market Data Incremental Refresh

6.5.4.1 FIX Message

Tag	Field Name		Req	Length	Description
Header					
1180	ApplID		Y	N/A	Identifier of the server sending the message.
1181	ApplSeqNum		N	I32	Sequence number of message on the Real-Time channel.
1021	MDBookType		N	I32	Indicates the type of service if the message is published to convey order book information. **Value** **Meaning** 2 Price Depth 3 Order Depth
262	MDReqID		N	N/A	Identifier of the Market Data Request this message relates to. Required if the message is disseminated via the Snapshot channel.
912	LastRptRequested		N	1	Indicates the last message sent in response to a retransmission request. **Value** **Meaning** Y Last Message
268	NoMDEntries		Y	I32	Number of market data entries in the message.
➡	279	MDUpdate Action	Y	1	Indicates the update type. **Value** **Meaning** 0 New 1 Change 2 Delete
➡	1173	MDSubBook Type	N	1	Sub book (Type of trading) the update relates to.Absence of this field should be interpreted as Regular (1). **Value** **Meaning** 1 Regular 4 EFP Trade

➡	269	MDEntryType	Y	1	The type of market data being published.

					Value	**Meaning**
					0	Bid
					1	Offer
					2	Trade
					3	Index Value
					4	Opening Price
					5	Closing Price
					7	High Price
					8	Low Price
					9	VWAP
					B	Volume
					C	Open Interest
					D	Spot Price (Spot price of the Commodity)
					K	UASR (Upper Active Spread Range)
					L	LASR (Lower Active Spread Range)
					N	Highest Bid Price
					O	Lowest Offer Price
					b	Market Bid
					c	Market Offer
					f	Previous Close
					g	Price Band
					h	TheoreticalPrice
					i	Delta
					j	Gamma
					k	Vega
					l	Theta
					m	Rho
					n	Opening Open Interest
					o	Closing Open Interest
					p	Change in Open Interest
					q	Day High Open Interest
					r	Day Low Open Interest
					s	Buy Order VWAP
					t	Sell Order VWAP
					u	Buy Order Qty
					v	Sell Order Qty

					w ImpliedVolatility x Turnover y Trades
➡	278	MDEntryID	N	13	Unique identifier of a market data entry. Required if MDEntryType (269) is Bid (0), Offer (1), Market Bid (b) or Market Offer (c), Trade (2) and the order book is published by order depth.
➡	55	Symbol	Y	30	Unique identifier of the instrument, index, market or sector.
➡	32027	PriceLimitEvent	N	1	Required if MDEntryType (269) is Price Band (g) **Value** **Meaning** 0 Breach 1 Relaxation 2 VWAP Computation 3 Other
➡	326	SecurityTradingStatus	N	2	The trading session the instrument will be moved into due to the price band breach Required if PriceLimitEvent (32027) is Breach (0) **Value** **Meaning** 2 Halt 17 Regular Trading
➡	32031	PBAffectedSide	N	1	Required if PriceLimitEvent (32027) is Breach (0) or Relaxation (1) **Value** **Meaning** 0 Upper 1 Lower
➡	32030	PriceBandLevel	N	(8,0)	Indicates the current PB level which is breached when PriceLimitEvent (32027) is Breach (0). Indicates the current PB level relaxed to when PriceLimitEvent (32027) is Relaxation (1) Will not be populated if the relaxation is due to a change in the Instrument Reference Price or First Day Price Band Parameter
➡	32029	PriceBandLimits	N	(30,10)	Required if MDEntryType (269) is Price Band (g) Value will be 2 if PriceLimitEvent (32027) is Breach. Value will be 1 if PriceLimitEvent (32027) is Relaxation (1), VWAP Computation (2) or Other (3)

➡	➡	32028	PriceLimitUpdate	Y	1	Communicates the current price band or the new price band Required if PriceBandLimits (32029) is specified **Value** **Meaning** 0 Current 1 New
➡	➡	32032	PriceBandValues	N	(30,10)	Required if PriceBandLimits (32029) is not zero Value will always be 2
➡	➡	➡	1306 PriceLimitType	Y	1	Describes how the price limits are expressed Required if PriceBandLimits (32029) is specified **Value** **Meaning** 0 Price 2 Percentage
➡	➡	➡	1148 LowLimitPrice	N	(8,0)	Value or percentage of the lower price band.
➡	➡	➡	1149 HighLimitPrice	N	(8,0)	Value or percentage of the upper price band.
➡	22009	InternalRefFlag		N	15	Communicates the Token ID of the instrument
➡	270	MDEntryPx		N	(8,0)	Price of the market data entry. Required if MDUpdateAction (279) is New (0) or Change (1) and MDEntryType (269) is not Imbalance (A), Volume (B), Market Bid (b), Market Offer (c), Buy Order Qty (u), Sell Order Qty (v) or Trades (y). Value may only be shown up to 8 decimals
➡	271	MDEntrySize		N	I32	Quantity of the market data entry. Required if MDUpdateAction (279) is New (0) or Change (1) and MDEntryType (269) is not Index Value (3), Opening Price (4), Closing Price (5), High Price (7), Low Price (8), VWAP (9), Highest Bid Price (N), Lowest Offer Price (O), Previous Close (f), Buy Order VWAP (s), Sell Order VWAP (t) or Turnover (x).

➡	272	MDEntryDate		N	9	Date the off-book trade being published was executed. The date will be specified in UTC and in the YYYYMMDD format. Required if MDEntry Type (269) is Spot Price (D).
➡	273	MDEntryTime		N	21	Time the trade being published was executed. The time will be specified in UTC and in the HH:MM:SS.sss format. Required if MDEntry Type (269) is Trade (2) or if MDEntry Type (269) is Spot Price (D).
➡	277	Trade Condition		N	2	Indicates the side of the order imbalance or the type of auction a trade was executed in. **Value** **Meaning** R Opening Price AJ Closing Price/Index ZZ Previous Closing Index
➡	346	Number OfOrders		N	I32	Number of orders represented in the aggregate quantity published for a bid or offer. Required if MDUpdateAction (279) is New (0) or Change (1), MDEntryType (269) is Bid (0), Offer (1), Market Bid (b) or Market Offer (c) and the order book is published by price depth.
➡	290	MDEntry PositionNo		N	I32	Display position of an order in the order book within a price level. Required if MDEntryType (269) is Bid (0), Offer (1), Market Bid (b) or Market Offer (c) and the order book is published by order depth.
➡	1023	MDPriceLevel		N	I32	Display position of a price level in the order book. Required if MDEntryType (269) is Bid (0) or Offer (1).
➡	1070	MDQuote Type		N	1	**Value** **Meaning** 0 Indicative
➡	83	RptSeq		N	I32	Instrument specific sequence number of update.
➡	453	NoPartyIDs		N	I32	Number of party identifiers. If specified, the value in this field will always be “1”.
➡	➡	448	Party ID	N	30	Identifier of the party. Required if NoPartyIDs (453) is specified.
➡	➡	447	Party ID Source	N	1	Required if PartyID (448) is specified. **Value** **Meaning** D Proprietary/Custom Code

➡	➡	452	Party Role	N	1	Role of the specified PartyID (448). Required if PartyID (448) is specified. **Value** **Meaning** 1 Trading Firm
➡	30002	OpenClose Indicator		N	2	Method used to compute the opening or closing price. Required if MDEntryType (269) is Opening Price (4) or Closing Price (5). **Value** **Meaning** 2 Regular Trade 3 Mid-Point 6 VWAP 7 Last Regular Trade 9 Previous Close 10 Manual 19 VWAP of Last n Trades 20 Reference Price 21 Price Unavailable 22 Best Bid 23 Best Offer 24 Theoretical Price
➡	31001	MDStatType		N	1	Whether the statistic is for a market or sector or an instrument type. **Value** **Meaning** 1 Market 2 Sector 3 Instrument Type 4 Instrument
➡	828	TrdType		N	2	Gives the type of trade carried out. Required if MDEntryType (269) is Trade (2) and MDSubBookType (1173) is All or None (5) **Value** **Meaning** 16 All or None
➡	1040	Secondary Trade ID		N	20	Numeric trade ID assigned for the trade. Required if MDEntry Type (269) is Trade(2)

6.5.5 News

6.5.5.1 FIX Message

Tag	Field Name		Req	Length	Description
Header					
1180	ApplID		Y	N/A	Identifier of the server sending the message.
1181	ApplSeqNum		N	I32	Sequence number of the message on the Real-Time channel. Required if the message is disseminated via the Real-Time or Replay channel.
42	OrigTime		Y	21	Time the announcement was published which will be specified in UTC and in the HH:MM:SS format.
61	Urgency		Y	I32	Level of urgency of the announcement. **Value** **Meaning** 0 Normal 1 Flash (High Priority) 2 Background (Low Priority)
148	Headline		Y	500	Headline or subject of the announcement.
912	LastRptRequested		N	1	Indicates the last message sent in response to a request. **Value** **Meaning** Y Last Message
33	NoLinesOfText		Y	I32	Number of lines of text. The value in this field will always be “1”.
➡	58	Text	Y	750	Text of the announcement.
146	NoRelatedSym		N	I32	Number of related instruments.
➡	55	Symbol	N	30	Unique identifier of the instrument. Required if NoRelatedSym (146) is specified.
711	NoUnderlyings		N	I32	Number of related underlying instruments.
➡	311	Underlying Symbol	N	30	Unique identifier of the underlying. Required if NoUnderlyings (711) is specified.

6.5.6 Market Data Request Reject

6.5.6.1 FIX Message

Tag	Field Name	Req	Length	Description
Header				
262	MDReqID	Y	N/A	Identifier of the request being rejected.
281	MDReqRej Reason	Y	61	Code specifying the reason for the rejection. Please refer to Section 9.1 for a list of reject codes.
58	Text	N	61	NCDEX specific code specifying the reason for the reject. Please refer to Section 9.1 for a list of reject codes.

6.5.7 Business Message Reject

6.5.7.1 FIX Message

Tag	Field Name	Req	Length	Description
Header				
379	BusinessReject RefID	N	N/A	SecurityReqID (320), MDReqID (262) or ApplReqID (1346) of the rejected message.
372	RefMsgType	N	2	MsgType (35) of the rejected message.
371	RefTagID	N	I32	If a message is rejected due to an issue with a particular field its tag number will be indicated.
380	BusinessReject Reason	Y	I32	Code specifying the reason for the reject. Please refer to Section 9.2 for a list of reject codes.
58	Text	N	61	NCDEX specific code specifying the reason for the reject. Please refer to Section 9.2 for a list of reject codes.

6.5.8 Application Message Request Ack

6.5.8.1 FIX Message

Tag	Field Name	Req	Length	Description
Header				
1353	ApplResponseID	Y	N/A	Server specified identifier of the acknowledgement.
1346	ApplReqID	Y	N/A	Identifier of the request being acknowledged.
1347	ApplReqType	Y	I32	Type of request being acknowledged. **Value** **Meaning** 0 Retransmission of Messages 5 Cancel Retransmission
1348	ApplResponse Type	Y	I32	Whether the request was successful. **Value** **Meaning** 0 Request Successful 1 Unknown ApplID 2 Messages Not Available

6.5.9 Application Message Report

6.5.9.1 FIX Message

Tag	Field Name	Req	Length	Description
Header				
1356	ApplReportID	Y	N/A	Server specified identifier of the report.
1346	ApplReqID	Y	N/A	Identifier of the Application Message Request the report relates to.
1426	ApplReportType	Y	I32	**Value** **Meaning** 3 Retransmission Completed

7 INSTRUMENT CLASSIFICATION

7.1 Segment

Segment	Description
Main	Main Board
Secondary	Secondary Board
Default	Default Board
FU	Futures
OP	Options
CS	Calendar Spreads

7.2 CFI Codes

CFI Code	Description
FXXXXX	*Future*
OXXXXX	*Option*
FMXXXX	*Futures Strategy*
OCXXXX	*Call Option*
OCXFXX	*Call Option on a Future*
OPXXXX	*Put Option*
OPXFXX	*Put Option on a Future*

7.3 Security Types

Security Type	Description
FUT	*Future*
MLEG	*Multi-Leg Instrument*
OOF	*Options on Futures*
OPT	*Option*

7.4 Security Sub Types

Security Sub Type	Description
CS	*Calendar Spread*

8 TRADING HALT REASON CODES

Code	Reason
100	Reason not available
101	Instrument-level circuit breaker tripped
102	Instrument status is Halted
200	Instrument price band breach
9998	Matching partition suspended
9999	System suspended

9 REJECT CODES

9.1 Market Data Request Reject

MDReq Rej Reason	Text	Reason
0	-	Unknown instrument
4	-	Unsupported SubscriptionRequestType
8	-	Unsupported MDEntryType
Z	-	Other
Z	101	Unknown segment
Z	102	Requested market data unavailable

9.2 Business Message Reject

Business Reject Reason	Text	Reason
0	400	Other
0	403	Incorrect data format for this tag
0	404	Value is invalid for this tag
0	405	Required tag missing
0	449	Concurrent request limit reached
0	450	Request limit for day reached
1	-	Unknown ID
3	-	Unsupported message type
5	-	Conditionally required field missing