



## **NSE – INDEX FEED**

**Version: 1.8**

**Date: March 08, 2016**



DOTEX INTERNATIONAL LIMITED  
EXCHANGE PLAZA,  
PLOT NO. C/1, G BLOCK,  
BANDRA-KURLA COMPLEX,  
BANDRA € , MUMBAI 400 051.  
INDIA.

## Revision History

<b>Name</b>	<b>Description</b>	<b>Date</b>
Version 1.0	New Specification Issued	October 16, 2012
Version 1.1	Correction in ST_COMP_BATCH_HEADER Point no 3.	November 30, 2012
Version 1.2	S&P is removed from the indices name Point no 10	February 12, 2013
Version 1.3	New Index addition "CPSE INDEX"	March 11, 2014
Version 1.4	New Index addition "NI15"	May 28, 2014
Version 1.5	Addition of 4 New Indices	September 30, 2014
Version 1.6	New Index addition "NV20"	June 12, 2015
Version 1.7	Index Name Rebranding	September 29, 2015
<b>Version 1.8</b>	<b>10 New Index Addition and Index Rename Change</b>	<b>March 08, 2016</b>

## Table of Contents

1.	Introduction .....	- 4 -
2.	Session Initialization .....	- 6 -
2.1	Structural Diagram .....	- 6 -
2.2	Online Requirements.....	- 7 -
2.3	Data Types.....	- 7 -
2.4	Acronyms Used .....	- 7 -
3.	Packet Format.....	- 8 -
3.1	Diagrammatic Representation of Packet Format:.....	- 9 -
4.	Session Messages.....	- 10 -
4.1	Login Request (Sent by client).....	- 10 -
4.2	Login Response (Sent by server).....	- 10 -
4.3	Heartbeat Message (Sent by server).....	- 11 -
5.	Sequenced Data Message (Sent by server).....	- 13 -
5.1	ONLINE - Market Status Message.....	- 13 -
5.2	ONLINE - Indices Information .....	- 14 -
6.	Steps For Decompressing The Data Packets .....	- 15 -
6.1	LZO Algorithm Details .....	- 15 -
6.2	Files required for LZO algorithm. ....	- 15 -
6.3	Decompression steps.....	- 15 -
7.	Checksum Calculation Algorithm.....	- 17 -
8.	Note .....	- 18 -
8.1	Normal Market Session .....	- 18 -
8.2	Auction Market Session.....	- 18 -
8.3	Pre-Open Session.....	- 18 -
8.4	Call Auction Session 1 .....	- 19 -
8.5	Call Auction Session 2 .....	- 19 -
9.	Support Information.....	- 20 -
10.	Annexure 1 .....	- 21 -

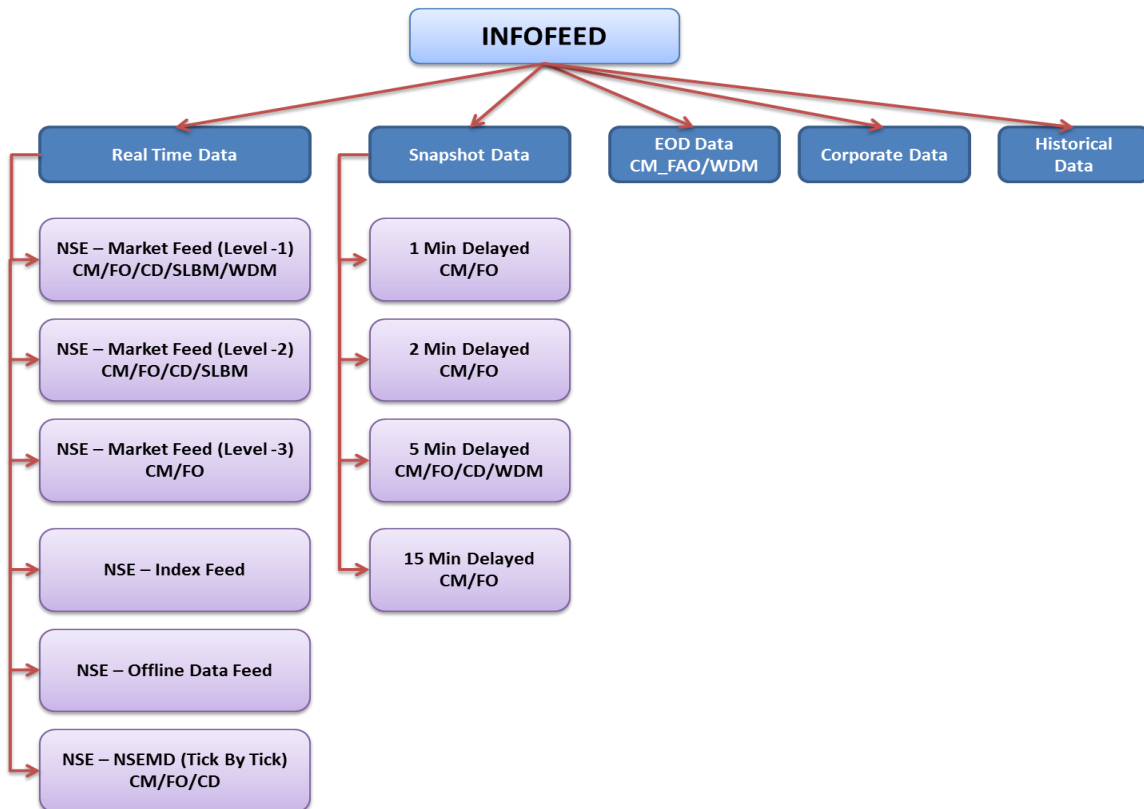
# NSE – Index Feed

## 1. Introduction

DotEx International Ltd. disseminates NSEIL’s real time broadcast data to various information agencies. It provides the 5 different types of data products to vendors.

- A. Real Time Data
- B. Snapshot Data
- C. End of Day Data
- D. Corporate Data
- E. Historical Data

The real time data and corporate data is a packet broadcast available in TCP/IP format, whereas the snapshot data, end of day data and historical data is available in the form of files. All these data products come under in Infofeed application.



In Infofeed's Real Time Data product following sub-products are available

- a. NSE - Market Feed (CM/FO/CD/SLBM/WDM Level 1)
- b. NSE - Market Feed (CM/FO/CD/SLBM Level 2)
- c. NSE - Market Feed (CM/FO Level 3)
- d. NSE - Index Feed
- e. NSE - Offline Data Feed
- f. NSE - NSEMD (CM/FO/CD Tick By Tick)

This document explains about the NSE – Index Feed product. Through this product on real time basis all the NSE's indices information is disseminated.

The information agencies connect to the Index Feed Server through Leased Lines. These leased lines are terminated on Infofeed Router and their data specific pneumatic calls are forwarded to Infofeed server. The Infofeed server accepts these pneumatic calls and creates a socket connection. The TCP/IP data flows to the information agencies through these socket connections.

The feed consist of series of sequenced and unsequenced variable length compressed messages. The compression algorithm used over here is LZO – Compression.

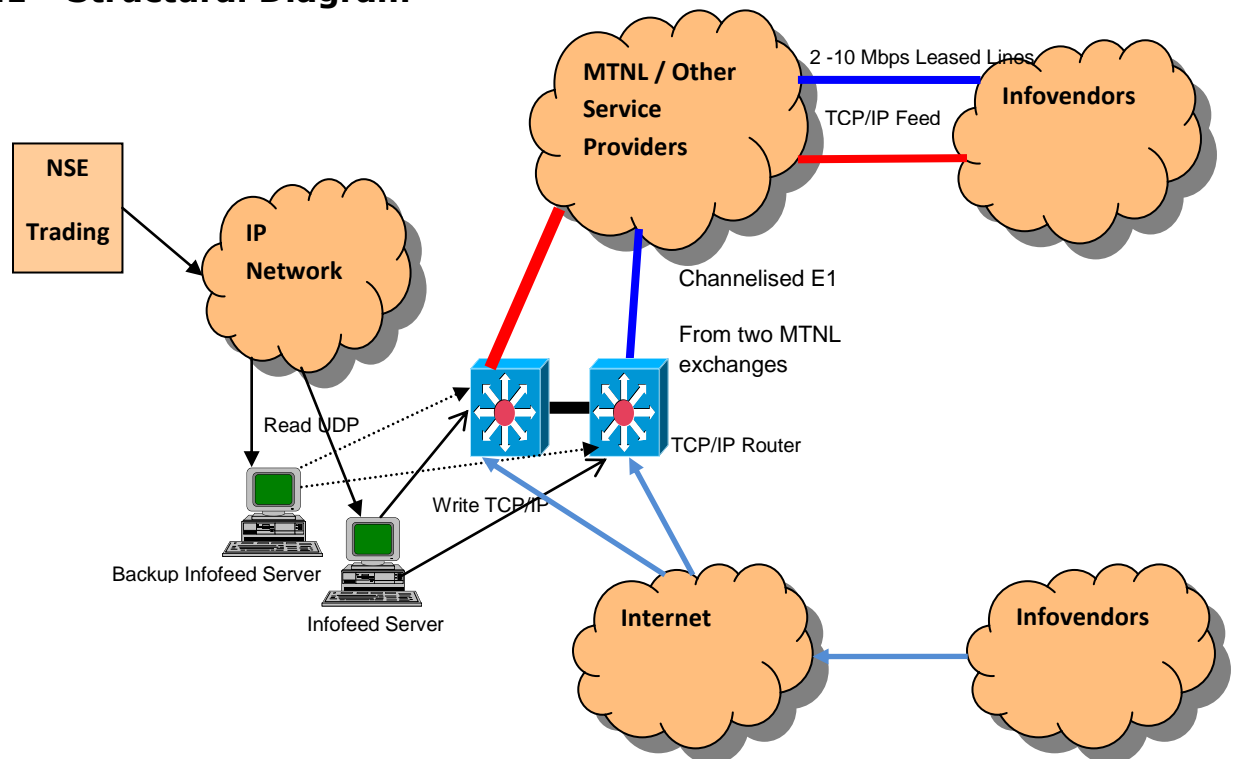
## 2. Session Initialization

NSE - Index Feed is built on TCP/IP socket connection. This feed consists of sequenced and unsequenced messages. Unsequenced messages provides the login and connection related messages such as login and heartbeats messages. Unsequenced messages are not part of the data. The sequenced data contains the actual market data and are reliable and recoverable.

A session begins with client establishing a TCP connection and sending the login request packet. Once the login request received the server authenticate it and send the login response. If the login is successful server will begin to send the sequenced data, or reject the login and terminate the connection.

Data packet consist of sequence number as one field. The first sequenced message of the day will send the sequence number as 1 and after that it will be incremented by 1 for each sequenced message. Client can recover the missed out data from separate NSE Offline Data system.

### 2.1 Structural Diagram



## 2.2 Online Requirements

- a) A Router / Switch or a card with TCP/IP capabilities to connect to 2 Mbps transmission lines for receiving NSEIL's Real time information.
- b) The Information agency should develop applications that initiate TCP/IP calls through 2 Mbps Leased Line.
- c) Information agency can connect to the Infofeed servers through the internet also. For IP validation at application level, information agencies has to provide the public static IP from which they will connect to Infofeed servers. Connectivity through internet is available for some products only.

## 2.3 Data Types

Data types used in feed,

Data Type	Size In Bytes
CHAR	1
INT	4
LONG	4
DOUBLE	8

Byte order - Little Endean.

## 2.4 Acronyms Used

ONLINE	Information Sent During Market Timing
CM	Cash Market
F&O	Future & Options Market
CD	Currency Derivatives Market
SLBM	Securities Lending & Borrowing Market
WDM	Whole Sale & Debt Market

### 3. Packet Format

Server sends all the packets in following format

typedef struct

```
{
    CHAR        cCompOrNot
    SHORT       nDataSize;
    SHORT       iNoOfPackets
}ST_COMP_BATCH_HEADER
```

typedef struct

```
{
    SHORT       iCode;
    SHORT       iLen;
    LONG        lSeqNo;
} ST_INFO_HEADER;
```

typedef struct

```
{
    .
    .
}ST_DATA_INFO;
```

typedef struct

```
{
    SHORT       iChecksum;
    CHAR        cEOT;
} ST_INFO_TRAILER;
```

typedef struct

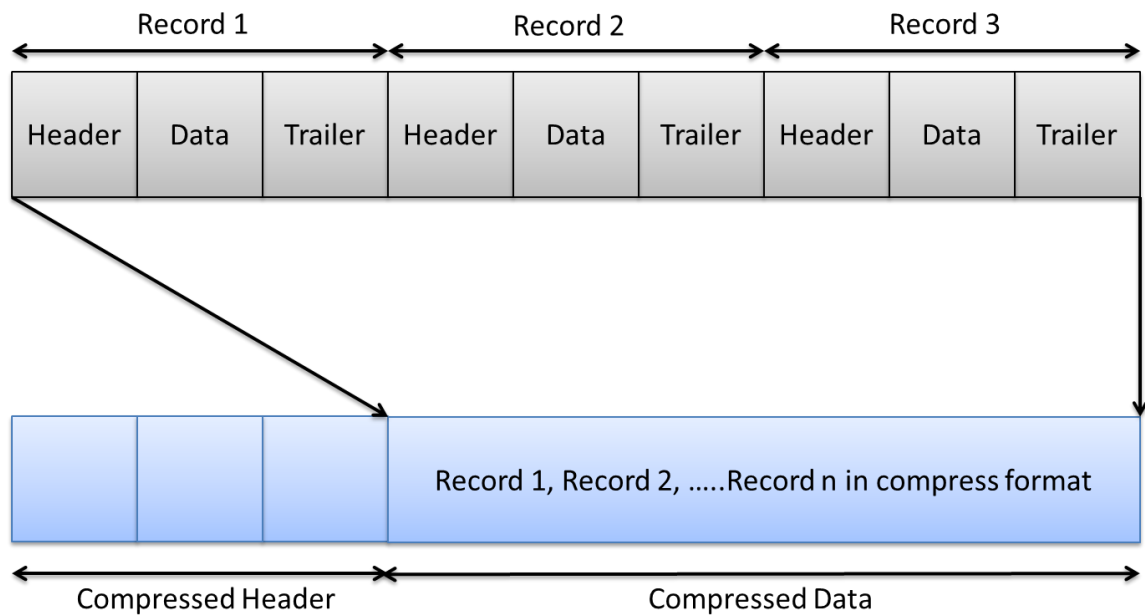
```
{
    ST_INFO_HEADER stInfoHdr;
    ST_DATA_INFO   stDataInfo;
    ST_INFO_TRAILER stInfoTrailer;
    .
}ST_DATA_PACKET
```

All the packets received from server consist of compress batch header. Compress batch header gives the information about the data packet compressed or not, number of packets in the following data packet and the



total size of data packet. Client needs to decompress the data packet using LZO decompression algorithm. After decompression each data packet consists of ST\_INFO\_HEADER, which has the iCode field to identify the type of the packet. Using iCode field, data info packet is mapped to the respective data packet.

### 3.1 Diagrammatic Representation of Packet Format:



#### Compressed Header

1. Compressed/ Uncompressed = 0 then compressed/ 1 uncompressed
2. Number of packets = Number of records in compressed data
3. Data Size = Compressed data size

As the data packets are sent in compressed format there is a need to decompress them. The compression algorithm used is LZO.

## 4. Session Messages

Session messages are not considered as market data messages. These messages provide the connection and login related messages such as login, and heartbeat messages.

### 4.1 Login Request (Sent by client)

Login request packet is sent by the client immediately after connecting to the server. This packet doesn't contain the compress batch header. If the client wants to change his default password then he needs to send "New Password" and "Confirm Password" in the request otherwise it should be kept blank. Password is case sensitive.

Field Name	Data Type	Value	Remark
<b>INFO HEADER</b>			
Code	SHORT	'CQ'	
Length	SHORT	Numeric	Size of (INFO HEADER + INFO DATA + INFO TRAILER)
Sequence Number	LONG	Numeric	0(Zero) for login request
<b>INFO DATA</b>			
User Id	CHAR[10]	Alphanumeric	Exchange provided user id (Null terminated)
Password	CHAR[8]	Alphanumeric	Password (Null terminated)
New Password	CHAR[8]	Alphanumeric	New password (Null terminated)
Confirm Password	CHAR[8]	Alphanumeric	Confirm password (Null terminated)
<b>INFO TRAILER</b>			
Check Sum	SHORT	Numeric	Refer point no. 7
End Of Trailer	CHAR	'\r'	Carriage Return

### 4.2 Login Response (Sent by server)

Login response packet will be sent by server after receiving the login request packet. This packet does contain the compress batch header.

This packet contains the error code from which the client can identify the status of the login.

Field Name	Data Type	Value	Remark
<b>INFO HEADER</b>			
Code	SHORT	'CR'	
Length	SHORT	Numeric	Size of (INFO HEADER + INFO DATA + INFO TRAILER)
Sequence Number	LONG	Numeric	0(Zero) for login response
<b>INFO DATA</b>			
Error Code	LONG		1000- Successful Login 1001- Password Update Successfully 1002- Wrong UserId- Password Combination 1003- Password is not valid in password change request. 1004- Login request is not correct. Error code other than above - Error in receiving logon response
Error Message	CHAR[50]		Description about the error code
<b>INFO TRAILER</b>			
Check Sum	SHORT	Numeric	Refer point no. 7
End Of Trailer	CHAR	'\r'	Carriage Return

### 4.3 Heartbeat Message (Sent by server)

Heartbeat message will be sent every 2 second if data is not available.

Field Name	Data Type	Value	Remark
<b>INFO HEADER</b>			
Code	SHORT	'CH'	
Length	SHORT	Numeric	Size of (INFO HEADER + INFO DATA + INFO TRAILER)
Sequence Number	LONG	Numeric	0(Zero) for heart beat message
<b>INFO DATA</b>			
Not associated with any data			

<b>INFO TRAILER</b>			
Check Sum	SHORT	Numeric	Refer point no. 7. Check sum is not calculated sent as 0(Zero),
End Of Trailer	CHAR	'\r'	Carriage Return

## 5. Sequenced Data Message (Sent by server)

Sequenced data messages will be sent by server and will contain the actual market data. These messages are reliable and recoverable as sequence number is assigned for each data message. For recovery please refer the NSE- Offline Data Feed technical specs

### 5.1 ONLINE - Market Status Message

This message is sent by the server, whenever the market status changes.

Field Name	Data Type	Value	Remark
<b>INFO HEADER</b>			
Code	SHORT	'PO' 'PC' 'CO' 'CC' 'CK' 'CL'	'PO' = Pre-open / Call Auction session start 'PC' = Pre-open / Call Auction session end 'CO' = Normal market open 'CC' = Normal market close 'CK' = Post close session start 'CL' = Post close session end
Length	SHORT	Numeric	Size of (INFO HEADER + INFO DATA + INFO TRAILER)
Sequence Number	LONG	Numeric	Application sequence number
<b>INFO DATA</b>			
Market Type	CHAR	Character	'N'-Normal 'S'- Spot 'O'- Odd Lot 'A'-Auction 'L'-All Markets 'C' – Call Auction 'G' – Reserved Market
<b>INFO TRAILER</b>			
Check Sum	SHORT	Numeric	Refer point no. 7. Check Sum not computed
End Of Trailer	CHAR	'\r'	Carriage Return

## 5.2 ONLINE - Indices Information

NSE-online indices information is sent through this message. For the list of the indices please refer the Annexure -1.

Field Name	Data Type	Value	Remark
<b>INFO HEADER</b>			
Code	SHORT	'CX'	
Length	SHORT	Numeric	Size of (INFO HEADER + INFO DATA + INFO TRAILER)
Sequence Number	LONG	Numeric	Application sequence number
<b>INFO DATA</b>			
Index Name	CHAR[17]	Character	Name of the Index
Current Index Value	CHAR[8]	Character	Current value of the Index. During pre-open session (i.e. between PO & PC msg with market type 'N') indicative index value is disseminated.
Open Index Value	CHAR[8]	Character	Current dates Opening value
Close Index Value	CHAR[8]	Character	Closing value of the Index. Before market close previous trading day's close value is sent.
High Index Value	CHAR[8]	Character	Current days high value of the index
Low Index Value	CHAR[8]	Character	Current days low value of the index
Percentage Change	CHAR[8]	Character	Percentage change in the index value
Yearly High Index Value	CHAR[8]	Character	Last 52 week high index value
Yearly Low Index Value	CHAR[8]	Character	Last 52 week low index value
<b>INFO TRAILER</b>			
Check Sum	SHORT	Numeric	Refer point no. 7.
End Of Trailer	CHAR	'\r'	Carriage Return

## **6. Steps For Decompressing The Data Packets**

### **6.1 LZO Algorithm Details**

LZO is a data compression library which is suitable for data de-/compression in real-time. This means it favors speed over compression ratio.

LZO is written in ANSI C. Both the source code and the compressed data format are designed to be portable across platforms.

LZO implements a number of algorithms with the following feature

- Decompression is simple and *\*very\** fast.
- Requires no memory for decompression.
- Requires 64 KB of memory for compression.
- Allows you to dial up extra compression at a speed cost in the compressor.
- The speed of the decompression is not reduced.
- Includes compression levels for generating pre-compressed data which achieve a quite competitive compression ratio.
- There is also a compression level which needs only 8 KB for Compression.
- Algorithm is thread safe.
- Algorithm is lossless.
- LZO supports overlapping compression and in-place decompression.

### **6.2 Files required for LZO algorithm.**

- Include files, source files (src) provided by LZO
- LZO.lib
- LZO library version used is 1.0.7

### **6.3 Decompression steps**

Receive the packet in the temporary buffer i.e. array of characters.

The first field is compressed or not compresses?

The second field is the number of packet in the following data packet.

The third field is data packet length.

Use the following function of LZO to Decompress.

```
r = lzo1z_decompress ((lzo_byte*)cInputBuf, ipLength,
(lzo_byte*)cOutputBuf, (lzo_uint*)&opLength, NULL);
```

**lzo1z\_decompress:** Function which decompresses the data packet received

**cInputBuf:** Input buffer in which compressed data is received

**ipLength:** The length of the packet which application has received using Receive ().

**cOutputBuf:** The uncompressed output data which is result of decompression.

**opLength:** Length of uncompressed data

After decompression data will be available in Output Buffer.

Each output data packet contains the INFO HEADER, after mapping the output decompressed buffer to INFO HEADER find out the data packet and the according to it map the output buffer to respective data packet.

**Algorithm:**

```
ST_NIFO_HEADER *pstInfoHeader;
for (i=0; i < iNoOfPackets; i++) // iNoOfPackets received in
                                // compressed data header
{
    pstInfoHeader = (ST_NIFO_HEADER *) cOutputBuf
    switch (pstInfoHeader->iCode)
    {
        case CX: //Indices Information
        {
            ST_INDEX_DATA *stIndexData = (ST_INDEX_DATA
            *)cOutputBuf;
            .
            .
            cOutputBuf = cOutputBuf + sizeof(ST_INDEX_DATA);
            break;
        }
    }
}
```



## 7. Checksum Calculation Algorithm

The Checksum routine followed for Info Vendor Feed is as follows:

// Following are the defines for checksum calculation

```

#define DC1      17
#define DC3      19
#define CR       13
#define LF       10
#define POLY     0x1021
// End of defines
unsigned check_sum (cData, iLength)
char *cData ;
int iLength;
{
    unsigned uAccum = 0;
    unsigned uData;
    unsigned char ucChk[2];
    int i,j;
    for (i=0;i<iLength;i++)
    {
        uData = *(cData+i);
        uData <= 8;
        for(j=8; j>0 ;j--){
            if((uData^uAccum)&0x8000)
                uAccum=(uAccum<<1)^POLY;
            /* SHIFT AND SUBTRACT POLY */
            else
                uAccum<<=1;
            uData<<=1;
        }
    }

    ucChk[0] = uAccum>>8;
    if (ucChk[0] == DC1 || ucChk[0] == DC3 || ucChk[0] == CR || ucChk[0] == LF )
        ucChk[0] -= 1;
    ucChk[1] = uAccum&0xFF;
    if (ucChk[1] == DC1 || ucChk[1] == DC3 || ucChk[1] == CR || ucChk[1] == LF )
        ucChk[1] -= 1;
    uAccum = ucChk[1];
    uAccum = (uAccum<<8) + ucChk[0];

    return(uAccum);
}

```

## **8. Note**

### **8.1 Normal Market Session**

All orders which are of regular lot size or multiples thereof are traded in the Normal Market. Normal market consists of various book types wherein orders are segregated as Regular lot orders, Special Term orders, Negotiated Trade Orders and Stop Loss orders depending on their order attributes.

### **8.2 Auction Market Session**

In the Auction Market, auctions are initiated by the Exchange on behalf of trading members for settlement related reasons. There are 3 participants in this market.

- Initiator - the party who initiates the auction process is called an initiator
- Competitor - the party who enters orders on the same side as of the initiator
- Solicitor - the party who enters orders on the opposite side as of the initiator

There is no impact on Index – Feed product because of this session

### **8.3 Pre-Open Session**

Pre-open session will be conducted for the Normal Market segment. The session will be conducted before the normal market start time. Exchange may decide to allow all or selective securities in pre-open session. During Pre-open session, only order entry, orders modification and order cancellation will be allowed. Once pre-open session ends, no order activity will be allowed and final open price (i.e. equilibrium price based on accumulated buy and sell orders) will be computed. Pre-open orders will be matched at this final open price resulting into trade execution. Pre-open orders that could not participate in the pre-open matching for the reasons such as demand-supply gap, order price worse than the equilibrium price etc. shall be carried forward to the normal market. The time priority of such orders shall be retained.

In the above context NSE – Index Feed product sends messages in following sequence

1. Pre-open session start (PO) – market type 'N'
2. Index Information (CX) – Indicative open index value in Current Index value
3. Security Update Information (PN) – Indicative open price in open price field
4. Pre-open session end (PC) - market type 'N'

5. Index Information (CX) – Derived final open price updated in open price field and current index value
6. Security Update Information (PN) – Derived final open price in open price field and current security information
7. Normal Market open (CO) - market type 'N'
8. Index Information (CX) – With current index information
9. Security Update Information (CN) – With current security
10. Normal Market Close (CC) - market type 'N'

#### **8.4 Call Auction Session 1**

SME (small and medium enterprises) securities call auction is done through this session. It is similar to the pre-open session. Multiple sessions of this can be held in a trading day. Market type for this session is 'C'

There is no impact on Index – Feed product because of this session

#### **8.5 Call Auction Session 2**

IPO/Relisting securities call auction is done through this session. It is similar to the pre-open session. Multiple sessions of this can be held in a trading day. Market type for this session is 'G'

There is no impact on Index – Feed product because of this session

## 9. Support Information

Name	Email	Contact Number
DOTEX Business	<b>dotex@nse.co.in</b>	91-22-26598385
Technical Support	<b>infofeed_support@nse.co.in.</b>	-

## 10. Annexure 1

List of indices available in NSE-Index Feed

Sr. No.	Index Name
1	NIFTY 50
2	NIFTY IT
3	NIFTY NEXT 50
4	NIFTY BANK
5	NIFTY MID100 FREE
6	NIFTY 500
7	NIFTY 100
8	NIFTY MIDCAP 50
9	NIFTY REALTY
10	NIFTY INFRA
11	INDIA VIX
12	NIFTY ENERGY
13	NIFTY FMCG
14	NIFTY MNC
15	NIFTY PHARMA
16	NIFTY PSE
17	NIFTY PSU BANK
18	NIFTY SERV SECTOR
19	NIFTY AUTO
20	NIFTY MEDIA
21	NIFTY METAL
22	NIFTY SML100 FREE
23	NIFTY 200
24	NIFTY DIV OPPS 50
25	NIFTY COMMODITIES
26	NIFTY CONSUMPTION
27	NIFTY FIN SERVICE
28	NIFTY50 DIV POINT
29	NIFTY100 LIQ 15
30	NIFTY CPSE
31	NIFTY GROWSECT 15
32	NIFTY50 TR 2X LEV
33	NIFTY50 PR 2X LEV
34	NIFTY50 TR 1X INV
35	NIFTY50 PR 1X INV
36	NIFTY50 VALUE 20

37	NIFTY QUALITY 30
38	NIFTY MID LIQ 15
39	NIFTY PVT BANK
40	NIFTY GS 8 13YR
41	NIFTY GS 10YR
42	NIFTY GS 10YR CLN
43	NIFTY GS 4 8YR
44	NIFTY GS 11 15YR
45	NIFTY GS 15YRPLUS
46	NIFTY GS COMPOSITE