



NSEMD Feed Specification  
(Equity Derivatives Market)

**Version:** 1.0

**Date:** November 30, 2009



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

**Revision History**

<b>Name</b>	<b>Description</b>	<b>Date</b>
Version 1.0	New Specification Issued	November 30, 2009

## **NSE – NSEMD Feed**

### **1. Introduction**

The NSE-MD Feed is the vendor level data feed of NSE India Limited. It disseminates information about orders and trades executed on NSE on real time basis. The feed consist of series of sequenced and unsequenced variable length compressed messages. The compression algorithm used over here is LZ0 – Compression.

NSEMD Feed data flows to the clients on TCP/IP sockets connection.

### **2. Session Initialization**

NSEMD 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, logout and heartbeats messages. Unsequenced messages are not part of the market 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 offline data server. For recovering the data client has to send a login request with valid start sequence number and the end sequence number. On receiving this login request on offline server, it will send the data from start sequence number till the end sequence number to that particular client. For example if client connection got disconnected and till that time he has received the last message having sequence number 50 and on reconnecting the first sequenced packet client received contain sequence number as 75. Then client can recover the missed out data by sending the login request to offline server, with start sequence number equal to 51 and end sequence number equal to 74.

### 3. Data Type

Data types used in NSEMD feed,

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

Byte order - Little Endean.

### 4. Packet Format

Server sends all the packets in following format

```
typedef struct
{
    int    iNoOfPackets;
    int    nDataSize;
}ST_COMP_BATCH_HEADER
```

```
typedef struct
{
    char  cMsgType;
    .
    .
    .
}ST_DATA_PACKET
```

All the packets received from server consist of compress batch header. Compress batch header gives the number of packets in the following data packet and the total size of data packet. Client needs to uncompress the data packet using LZ0 compression algorithm. Each data packet consists of message type as the first field from which the type of the packet can be identified according to it packet is mapped to the data packets.

## 5. Session Messages

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

### 5.1. 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" in the request otherwise it should be kept blank. Password is case sensitive.

For recovery of messages, client should send the "Start Sequence Number" and "End Sequence Number" in the request. To get on-line messages client should send zero (0) in both "Start Sequence Number" and "End Sequence Number".

Field Name	Data Type	Value	Remark
Message Type	CHAR	'L'	Login Request
Username	CHAR [10]	Alphanumeric	Username
Password	CHAR [8]	Alphanumeric	Password
New Password	CHAR [8]	Alphanumeric	New Password
Start Sequence Number	LONG	Numeric	Start sequence number in case of recovery
End Sequence Number	LONG	Numeric	End sequence number in case of recovery

### 5.2. Login Response (Sent by server)

Login response packet will be sent by server after receiving the login request packet. This packet doesn't 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
Message Type	CHAR	'R'	Login Response
Error Code	INT	Numeric	1000 = Login Successful 1001 = Login Successful.

			Password Updated Successfully. 1002 = Wrong Username Password Combination. 1003 = Password is not valid in password change request. 1004 = Login request is not correct. 1005 = Client is disabled 1006 = Offline client is already connected
--	--	--	--

**5.3. Logout Request (Sent by client)**

Logout request is sent by the client to terminate the session from the server. This packet doesn't contain the compress batch header.

Field Name	Data Type	Value	Remark
Message Type	CHAR	'O'	Logout Request

**5.4. Heartbeat Message (Sent by server)**

Heartbeat message will be sent every 1 second if data is not available.

Field Name	Data Type	Value	Remark
Message Type	CHAR	'H'	Heartbeat Message

## 6. 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. These messages will contain normal market and regular lot data.

### 6.1. Market Status Message

Whenever the market status changed this message is sent by the server.

Field Name	Data Type	Value	Remark
Message Type	CHAR	'S'	Market Status Message
Sequence Number	LONG	Numeric	Message Sequence Number
Market Type	CHAR	Character	'N' – Normal Market
Market Status	CHAR	Character	'O' – Open 'C' – Close

### 6.2. Order Message

Any new order accepted and added to the order book or any order cancelled then this message is sent.

Field Name	Data Type	Value	Remark
Message Type	CHAR	'N' or 'X'	'N' - New Order 'X' - Order Cancellation
Sequence Number	LONG	Numeric	Message Sequence Number
Timestamp	LONG	Numeric	Time in milliseconds from 01-Jan-1980 00:00:00
Instrument Type	CHAR	Numeric	'1' - FUTIDX '2' - FUTSTK '3' - OPTIDX '4' - OPTSTK
Symbol	CHAR[10]	Alphanumeric	Security Symbol. The symbol received is right padded with blanks and is not NULL terminated.
Expiry Date	INT	Numeric	Expiry date of contract in seconds from 01-Jan-1980 00:00:00

Strike Price	INT	Numeric	Strike Price Of Contract (In Paise). This should be divided by 100 for converting into Rupees. This will be zero in case of futures contract.
Option Type	CHAR	Alphanumeric	'0' - For futures contract '1' - CA '2' - PA '3' - CE '4' - PE
Order Type	CHAR	Character	'B' - Buy Order 'S' - Sell Order
Price	INT	Numeric	Price of the order (In Paise) This field contains the price at which the order is placed. The price is in multiples of the tick size. This should be divided by 100 for converting into Rupees.
Quantity	INT	Numeric	Quantity of the order

### 6.3. Order Modification Message

Any order modified and added to order book then this message is sent. It contains the new and old image (i.e. price and quantity) of the order.

Field Name	Data Type	Value	Remark
Message Type	CHAR	Character	'M' - Order Modification Message
Sequence Number	LONG	Numeric	Message Sequence Number
Timestamp	LONG	Numeric	Time in milliseconds from 01-Jan-1980 00:00:00
Instrument Type	CHAR	Numeric	'1' - FUTIDX '2' - FUTSTK '3' - OPTIDX '4' - OPTSTK
Symbol	CHAR[10]	Alphanumeric	Security Symbol. The symbol received is right

			padding with blanks and is not NULL terminated.
Expiry Date	INT	Numeric	Expiry date of contract in seconds from 01-Jan-1980 00:00:00
Strike Price	INT	Numeric	Strike Price Of Contract (In Paise). This should be divided by 100 for converting into Rupees. This will be zero in case of futures contract.
Option Type	CHAR	Alphanumeric	'0' – For futures contract '1' - CA '2' - PA '3' - CE '4' – PE
Order Type	CHAR	Character	'B' – Buy Order 'S' – Sell Order
New Price	INT	Numeric	Modified Price of the order (In Paise) This field contains new price of the order. The price is in multiples of the tick size. This should be divided by 100 for converting into Rupees.
New Quantity	INT	Numeric	Modified Quantity of the order
Old Price	INT	Numeric	Old Price of the order (In Paise) This field contains the price which was modified. The price is in multiples of the tick size. This should be divided by 100 for converting into Rupees.
Old Quantity	INT	Numeric	Old Quantity of the order

#### 6.4. Trade Message

This message is sent whenever an order in the order book gets executed fully or partially.

Field Name	Data Type	Value	Remark
Message Type	CHAR	Character	'T' = Trade Message 'I' = Hidden Quantity Trade
Sequence Number	LONG	Numeric	Message Sequence Number
Timestamp	LONG	Numeric	Time in milliseconds from 01-Jan-1980 00:00:00
Instrument Type	CHAR	Numeric	'1' - FUTIDX '2' - FUTSTK '3' - OPTIDX '4' - OPTSTK
Symbol	CHAR[10]	Alphanumeric	Security Symbol. The symbol received is right padded with blanks and is not NULL terminated.
Expiry Date	INT	Numeric	Expiry date of contract in seconds from 01-Jan-1980 00:00:00
Strike Price	INT	Numeric	Strike Price Of Contract (In Paise). This should be divided by 100 for converting into Rupees. This will be zero in case of futures contract.
Option Type	CHAR	Alphanumeric	'0' – For futures contract '1' - CA '2' - PA '3' - CE '4' – PE
Trade Price	INT	Numeric	Trade Price (In Paise) This field contains the price at which the trade took place. The price is in multiples of the tick size.

			This should be divided by 100 for converting into Rupees
Trade Quantity	INT	Numeric	Trade Quantity
Open Price	INT	Numeric	Day Open Price of the Security (In Paise) This field contains the open price of a security. The price is in multiples of the tick size. This should be divided by 100 for converting into Rupees
High Price	INT	Numeric	High Price of the Security (In Paise) This field contains the high price of a security till that point of time. The price is in multiples of the tick size. This should be divided by 100 for converting into Rupees
Low Price	INT	Numeric	Low Price of the Security This field contains the low price of a security till that point of time. The price is in multiples of the tick size. This should be divided by 100 for converting into Rupees
Total Turn Over	DOUBLE	Numeric	Total Turn Over of security till that point of time

## 7. Examples

### 7.1. Passive Order Added To Book

Action	Type	NSEMD Message
Sell order of 50 for contract "OPTSTK NTPC DEC09 220.00 CA" is entered at price 50.50	Passive	Message Type = 'N' Sequence Number = 234 Time Stamp = 945252600000 Instrument Type = '4' Symbol = "NTPC " Expiry Date = 946684800 Strike Price = 22000 Option Type = '1' Order Type = 'S' Price = 5050 Quantity = 50

### 7.2. Active Order Gets Fully Traded

Action	Type	NSEMD Message
Sell order of 50 for contract "OPTSTK NTPC DEC09 220.00 CA" is entered at price 50.50	Passive	Message Type = 'N' Sequence Number = 234 Time Stamp = 945252600000 Instrument Type = '4' Symbol = "NTPC " Expiry Date = 946684800 Strike Price = 22000 Option Type = '1' Order Type = 'S' Price = 5050 Quantity = 50
Matching buy order entered	Trade	Message Type = 'T' Sequence Number = 240 Time Stamp = 945252600700 Instrument Type = '4' Symbol = "NTPC " Expiry Date = 946684800 Strike Price = 22000 Trade Price = 5055

		Trade Quantity = 50 Open Price = 4550 High Price = 5500 Low Price = 4200 Total Turnover = 1284597876
--	--	--

### 7.3. Active Order Partially Traded

Action	Type	NSEMD Message
Buy order of 111 for contract "OPTSTK NTPC DEC09 220.00 CA" is entered at price 50.50	Passive	Message Type = 'N' Sequence Number = 234 Time Stamp = 945252600000 Instrument Type = '4' Symbol = "NTPC " Expiry Date = 946684800 Strike Price = 22000 Option Type = '1' Order Type = 'B' Price = 5050 Quantity = 111
Matching sell order entered of 112 quantity.	Trade	Message Type = 'T' Sequence Number = 240 Time Stamp = 945252600700 Instrument Type = '4' Symbol = "NTPC " Expiry Date = 946684800 Strike Price = 22000 Option Type = '1' Trade Price = 5050 Trade Quantity = 111 Open Price = 4550 High Price = 5500 Low Price = 4200 Total Turnover = 1284597876
Addition of residual 1 quantity of sell order to book	Passive	Message Type = 'N' Sequence Number = 241 Time Stamp = 945252600700 Instrument Type = '4' Symbol = "NTPC "

		Expiry Date = 946684800 Strike Price = 22000 Option Type = '1' Order Type = 'S' Price = 5050 Quantity = 1
--	--	--

**7.4. Market Order Added To Book**

Action	Type	NSEMD Message
Sell order of 111 for contract "OPTSTK NTPC DEC09 220.00 CA" is entered as MKT order. (Market Price = 50.50)	Passive	Message Type = 'N' Sequence Number = 247 Time Stamp = 945252600000 Instrument Type = '4' Symbol = "NTPC " Expiry Date = 946684800 Strike Price = 22000 Option Type = '1' Order Type = 'S' Price = 5050 Quantity = 111

**7.5. Market Order Gets Fully Traded**

Action	Type	NSEMD Message
Sell order of 111 for contract "OPTSTK NTPC DEC09 220.00 CA" is entered as MKT order and gets fully traded (Market Price = 50.50)	Active	Message Type = 'T' Sequence Number = 240 Time Stamp = 945252600000 Instrument Type = '4' Symbol = "NTPC " Expiry Date = 946684800 Strike Price = 22000 Option Type = '1' Trade Price = 5050 Trade Quantity = 111 Open Price = 4550 High Price = 5500 Low Price = 4200 Total Turnover = 1284597876

**7.6. Market Order Partially Traded**

<b>Action</b>	<b>Type</b>	<b>NSEMD Message</b>
Sell order of 100 for contract "OPTSTK NTPC DEC09 220.00 CA" is entered as MKT order and trade result for 50 shares (Market Price = 50.50)	Trade	Message Type = 'T' Sequence Number = 240 Time Stamp = 945252600000 Instrument Type = '4' Symbol = "NTPC" Expiry Date = 946684800 Strike Price = 22000 Option Type = '1' Traded Price = 5050 Traded Quantity = 50 Open Price = 4550 High Price = 5500 Low Price = 4200 Total Turnover = 1284597876
Addition of residual 50 quantity of sell order to book at "MKT" price. (Market Price = 50.50)	Passive	Message Type = 'N' Sequence Number = 241 Time Stamp = 945252600000 Instrument Type = '4' Symbol = "NTPC" Expiry Date = 946684800 Strike Price = 22000 Option Type = '1' Order Type = 'S' Price = 5050 Quantity = 50

**7.7. Order Modification (Price)**

<b>Action</b>	<b>Type</b>	<b>NSEMD Message</b>
Buy order of 111 for contract "OPTSTK NTPC DEC09 220.00 CA" is entered at price 50.50	Passive	Message Type = 'N' Sequence Number = 242 Time Stamp = 945252600000 Instrument Type = '4' Symbol = "NTPC" Expiry Date = 946684800 Strike Price = 22000

		Option Type = '1' Order Type = 'B' Price = 5050 Quantity = 111
Price modified to 51.00	Passive	Message Type = 'M' Sequence Number = 254 Time Stamp = 945252600800 Instrument Type = '4' Symbol = "NTPC " Expiry Date = 946684800 Strike Price = 22000 Option Type = '1' Order Type = 'B' New Price = 5100 New Quantity = 111 Old Price = 50.50 Old Quantity = 111

**7.8. Order Modification(Price) Results In Trade**

Action	Type	NSEMD Message
Sell order of 111 for contract "OPTSTK NTPC DEC09 220.00 CA" is entered at price 50.50	Passive	Message Type = 'N' Sequence Number = 252 Time Stamp = 945252600000 Instrument Type = '4' Symbol = "NTPC " Expiry Date = 946684800 Strike Price = 22000 Option Type = '1' Order Type = 'S' Price = 5050 Quantity = 111
Price modified to 51.00 results in trade	Order Cancellation	Message Type = 'X' Sequence Number = 253 Time Stamp = 945252600800 Instrument Type = '4' Symbol = "NTPC " Expiry Date = 946684800 Strike Price = 22000

		Option Type = '1' Order Type = 'S' Price = 5050 Quantity = 111
Trade result at price 51.00	Trade	Message Type = 'T' Sequence Number = 254 Time Stamp = 945252600800 Instrument Type = '4' Symbol = "NTPC " Expiry Date = 946684800 Strike Price = 22000 Option Type = '1' Traded Price = 5100 Traded Quantity = 50 Open Price = 4550 High Price = 5500 Low Price = 4200 Total Turnover = 1284597876

**7.9. Order Modification (Quantity)**

Action	Type	NSEMD Message
Sell order of 111 for contract "OPTSTK NTPC DEC09 220.00 CA" is entered at price 50.50	Passive	Message Type = 'N' Sequence Number = 252 Time Stamp = 945252600000 Instrument Type = '4' Symbol = "NTPC " Expiry Date = 946684800 Strike Price = 22000 Option Type = '1' Order Type = 'S' Price = 5050 Quantity = 111
Quantity modified to 110	Passive	Message Type = 'M' Sequence Number = 262 Time Stamp = 945252600800 Instrument Type = '4' Symbol = "NTPC " Expiry Date = 946684800

		Strike Price = 22000 Option Type = '1' Order Type = 'S' New Price = 5050 New Quantity = 110 Old Price = 5050 Old Quantity = 111
--	--	---

**7.10. Order Cancellation**

Action	Type	NSEMD Message
Sell order of 100 cancelled for contract "OPTSTK NTPC DEC09 220.00 CA" at price 50.50	Passive	Message Type = 'X' Sequence Number = 272 Time Stamp = 945252600000 Instrument Type = '4' Symbol = "NTPC" Expiry Date = 946684800 Strike Price = 22000 Option Type = '1' Order Type = 'S' Price = 5050 Quantity = 100

## 8. Steps For Decompressing The Data Packets

### 8.1. 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.

### 8.2. Files required for LZO algorithm.

- Include files, source files (src) provided by LZO
- LZO.lib
- LZO library version used is 2.03

### 8.3. Decompression steps

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

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

The second field is data packet length.

Use the following function of LZO to Decompress.

```
r = lzo1x_decompress_safe ((lzo_byte*)cInputBuf, ipLength,
(lzo_byte*)cOutputBuf, (lzo_uint*)&opLength, NULL);
```

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

**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 data packet contains the Message Type as the first field, read that field and according to that map the output buffer to data packets

**Algorithm:**

```
char *cTemp = cOutputBuf;
for (i=0; i < iNoOfPackets; i++)
{
    char *cMessageType = cTemp;
    switch(*cMessageType)
    {
        case ORDER_MESSAGE:
        {
            ST_ORDER_DATA *stOrderData = cTemp;
            .
            .
            cTemp = cTemp + sizeof(ST_ORDER_DATA);
            break;
        }
        case TRADE_MESSAGE:
        {
            ST_TRADE_DATA *stTradeData = cTemp;
            .
            .
            cTemp = cTemp + sizeof(ST_TRADE_DATA);
            break;
        }
    }
}
```

## 9. Support Information

For any further assistance please find below the support details.

<b>Email Address</b>
infofeed_support@nse.co.in