MARKET FEED
Wholesale Debt Market (WDM)
(LEVEL – 1)

Version: 1.3

Date: 27 October 2025

NSE DATA & ANALYTICS LIMITED
EXCHANGE PLAZA,
PLOT NO. C/1, G BLOCK,
BANDRA-KURLA COMPLEX,
BANDRA (E), MUMBAI 400 051.
INDIA.

COPYRIGHT NOTICE

Revision History

Name	Description	Date
Version 1.0	New Specification Issued	24 April 2009
Version 1.1	Removal of TCP/IP Session, Login Request, Login Response.	29 October 2021
Version 1.2	Change in Security Status from Suspended – 'S' to Participation – 'P' in GLOSSARY section	03 July 2023
Version 1.3	Addition of FAQs Section	27 October 2025

Table of Contents

Market Feed – Wholesale Debt Market (Level – 1)

1 Introduction

NSE Data & Analytics Ltd. offers real-time data and historical data products from NSEIL to a diverse range of clients. This includes 5 real-time products and historical data products:

Real Time data products

1. Real Time Data
2. Snapshot Data
3. Corporate Data
4. Analytical Products data
5. Indicative NAV Data

Historical data products

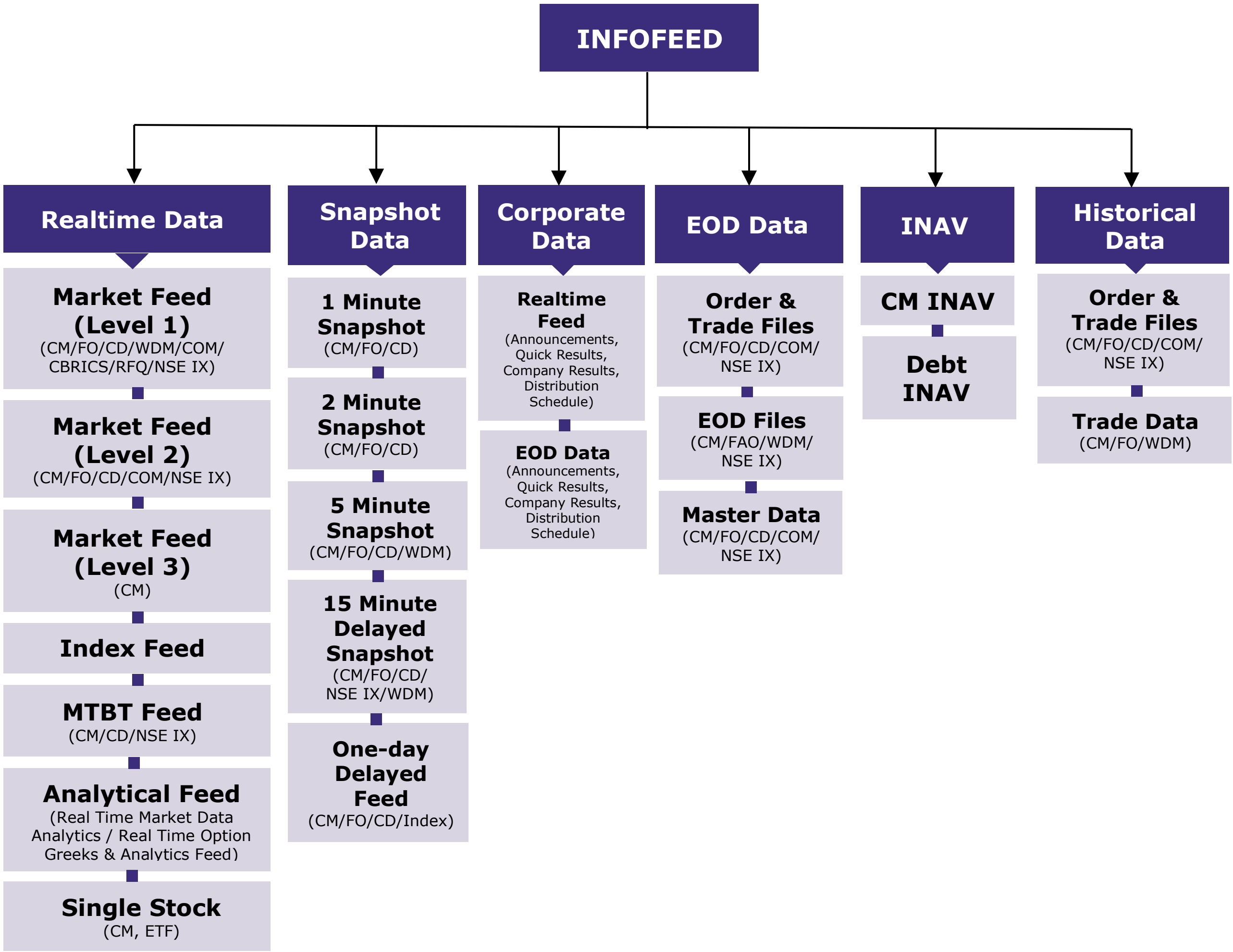
1. End of Day Data
2. Historical Data

The data products are provided through delivery modes mentioned below:

- **Real-time Data**: The information is transmitted as a packet broadcast, facilitating ongoing distribution through data feeds via point-to-point leased line.

- **Snapshot, End-of-Day, and Historical Data**: The data is delivered as downloadable files over the internet using the SFTP protocol.

All these data categories are integrated within the Infofeed platform, ensuring comprehensive coverage and streamlined access.


INFOFEED
Realtime Data
Market Feed (Level 1) (CM/FO/CD/WDM/COM/ CBRICS/RFQ/NSE IX)
Market Feed (Level 2) (CM/FO/CD/COM/NSE IX)
Market Feed (Level 3) (CM)
Index Feed
MTBT Feed (CM/CD/NSE IX)
Analytical Feed (Real Time Market Data Analytics / Real Time Option Greeks & Analytics Feed)
Single Stock (CM, ETF)
Snapshot Data
1 Minute Snapshot (CM/FO/CD)
2 Minute Snapshot (CM/FO/CD)
5 Minute Snapshot (CM/FO/CD/WDM)
15 Minute Delayed Snapshot (CM/FO/CD/ NSE IX/WDM)
One-day Delayed Feed (CM/FO/CD/Index)
Corporate Data
Realtime Feed (Announcements, Quick Results, Company Results, Distribution Schedule)
EOD Data (Announcements, Quick Results, Company Results, Distribution Schedule)
EOD Data
Order & Trade Files (CM/FO/CD/COM/ NSE IX)
EOD Files (CM/FAO/WDM/ NSE IX)
Master Data (CM/FO/CD/COM/ NSE IX)
INAV
CM INAV
Debt INAV
Historical Data
Order & Trade Files (CM/FO/CD/COM/ NSE IX)
Trade Data (CM/FO/WDM)

This document explains about the NSE – Market Feed (WDM Level 1) products. Through this product on a real-time basis all the NSE's market update information is disseminated.

The information agencies connect to the Market Feed Server through Leased Lines. These leased lines are terminated on Infofeed Router, and their data specific pneumonic calls are forwarded to Infofeed server.

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

2 Packet Format

Server sends all the packets in the 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 the server consist of a compress batch header. The compress batch header gives information whether the data packets are 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 packets are mapped to the respective data packets.

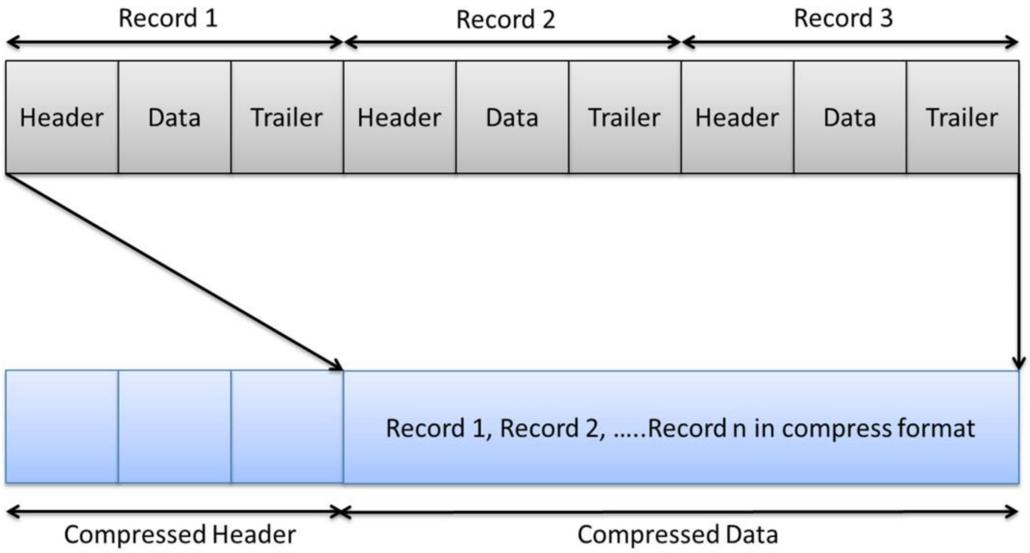
2.1 Data Types

Data types used in feed:

Data Type	Size In Bytes
CHAR	1
SHORT	2
LONG	4
DOUBLE	8

Byte order - Big Endian

2.2 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.

3 Data Details

A single data packet contains Header, the ASCII block of data and trailer. The interpretation of the ASCII data depends on the code field in the Header.

The Header:
The header consists of

Code Field (2 bytes) –
The code field indicates the type of data packet. The different types of code fields are - New Update (WN), Login Request (WQ), Login Response (WR), Market Open (WO), Market Close (WC), and End of day Market Status (WS), End of Feed (WE), and Heartbeat Signal (WH).

Length Field (2 bytes) –
The length field indicates the length of the packet that is being sent by the system. This field is computed in Hexadecimal format. This length includes the length of the header, trailer and the carriage return.

Sequence Number Field (4 bytes) –
The sequence number field indicates the sequence number of the packet that is being sent by the system. This is a unique number generated for a single trading for a market segment.

ASCII Data Block
The data block contains the actual data in ASCII format. The types of data block provided by the Exchange has been detailed –

a) Market Open
b) Market Close
c) Trade information (New update)
d) Market Status at the end of the day
e) End of the feed
f) Heartbeat Signal

Trailer
The trailer is a two-byte checksum. A CR terminates the block of data.

4 Data Structure Details

4.1 Market Open

The Market Open packet is sent when the WDM segment is opened for trading for that trading day. This packet is sent once a day. The WDM Market opens at 10:00 AM in the morning on each trading day.

Field Name	Data Type	Value	Brief Description
INFO HEADER			
Code	SHORT	‘WO’	‘WO’ = Market Open
Length	SHORT	Numeric	Size of (INFO HEADER + INFO DATA + INFO TRAILER)
Sequence Number	LONG	Numeric	Application sequence number
INFO DATA			
Message	CHAR [100]	Character	Message
INFO TRAILER			
Checksum	SHORT	Numeric	Refer to section checksum calculation Checksum is not calculated, so it is sent as 0(Zero)
End Of Trailer	CHAR [1]	‘\r’	Carriage Return

4.2 Market Close

The market close packet is sent when WDM segment is closed for that trading. The WDM segment is divided into 2 types, i.e., the same day settlement and other day settlement. The market close timing for both markets is different. The same day settlement market closes at 2:30 PM from Monday to Friday while on Saturday it closes at 1:00 PM. Similarly, the other day settlement closes at 05:30 PM from Monday to Friday while on Saturday it closes at 1:00 PM. Hence market close packets are sent twice a day in WDM segment.

Field Name	Data Type	Value	Brief Description
INFO HEADER			
Code	SHORT	'WC'	'WC' = Market Close
Length	SHORT	Numeric	Size of (INFO HEADER + INFO DATA + INFO TRAILER)
Sequence Number	LONG	Numeric	Application sequence number
INFO DATA			
Message	CHAR [100]	Character	Message
INFO TRAILER			
Checksum	SHORT	Numeric	Refer to section checksum calculation Checksum is not calculated, so it is sent as 0(Zero)
End Of Trailer	CHAR [1]	'\r'	Carriage Return

4.3 Trade Information (New Update)

This packet is sent during the trading hours

Field Name	Data Type	Value	Brief Description
INFO HEADER			
Code	SHORT	‘WN’	‘WN’ = Normal market updates
Length	SHORT	Numeric	Size of (INFO HEADER + INFO DATA + INFO TRAILER)
Sequence Number	LONG	Numeric	Application sequence number
INFO DATA			
Security Type	CHAR [2]	Character	Security Type
Security Name	CHAR [7]	Character	Security Name
Issue Name	CHAR [6]	Character	Issue Name
Settlement Days	CHAR [3]	Character	Settlement Days
Trade Type	CHAR [2]	Character	Trade Type
Repo Term	CHAR [3]	Character	Repo Term
Trade High Price	CHAR [10]	Character	Trade High Price
Trade Low Price	CHAR [10]	Character	Trade Low Price
Last Traded Price	CHAR [10]	Character	Last Traded Price
Total Traded Value	CHAR [10]	Character	Total Traded Value
Security Status	CHAR [1]	Character	Security Status
INFO TRAILER			
Checksum	SHORT	Numeric	Refer to section checksum calculation
End Of Trailer	CHAR [1]	‘\r’	Carriage Return

4.4 Heartbeat Data

Heartbeat is a packet that is sent at regular intervals throughout the day to communicate that the Market environment is up.

Field Name	Data Type	Value	Brief Description
INFO HEADER			
Code	SHORT	‘WH’	
Length	SHORT	Numeric	Size of (INFO HEADER + INFO TRAILER)
Sequence Number	LONG	Numeric	0(Zero) for heartbeat message
INFO DATA			
Not associated with any data			
INFO TRAILER			
Checksum	SHORT	Numeric	Refer to section checksum calculation Checksum is not calculated, so it is sent as 0 (Zero)
End Of Trailer	CHAR [1]	‘\r’	Carriage Return

4.5 Market Statistics Data (EOD Data)

The market status data contains the closing information of the securities that have been traded for that day. This feed is sent approximately at 6:40 p.m. on all Trading Days.

Field Name	Data Type	Value	Brief Description
INFO HEADER			
Code	SHORT	'WS'	
Length	SHORT	Numeric	Size of (INFO HEADER + INFO DATA + INFO TRAILER) (Variable length depending upon Message Length field of INFO DATA structure)
Sequence Number	LONG	Numeric	Application sequence number
INFO DATA			
Security Type	CHAR [2]	Character	Security Type
Security Name	CHAR [7]	Character	Security Name
Issue Name	CHAR [6]	Character	Issue Name
Settlement Days	CHAR [3]	Character	Settlement Days
Trade Type	CHAR [2]	Character	Trade Type
Repo Term	CHAR [3]	Character	Repo Term
Trade High Price	CHAR [10]	Character	Trade High Price
Trade Low Price	CHAR [10]	Character	Trade Low Price
Last Traded Price	CHAR [10]	Character	Last Traded Price
Total Traded Value	CHAR [10]	Character	Total Traded Value
Weighted Yield	CHAR [8]	Character	Weighted Yield
INFO TRAILER			
Checksum	SHORT	Numeric	Refer to section checksum calculation
End Of Trailer	CHAR [1]	'\r'	Carriage Return

4.6 End of the Feed

This packet signals that all the parts of End of the Day feed have been completed.

Field Name	Data Type	Value	Brief Description
INFO HEADER			
Code	SHORT	'WE'	
Length	SHORT	Numeric	Size of (INFO HEADER + INFO TRAILER)
Sequence Number	LONG	Numeric	Application sequence number
INFO DATA			
Not associated with any data			
INFO TRAILER			
Checksum	SHORT	Numeric	Refer to section checksum calculation Checksum is not calculated, so it is sent as 0 (Zero)
End Of Trailer	CHAR [1]	'\r'	Carriage Return

5 Steps for Decompressing the Data Packets

5.1 LZO Algorithm Details

The LZO stands for Lempel Ziv Oberhaumer. It is a data compression library which is suitable for data Decompression 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. This algorithm is freely available on the internet (URL: https://www.oberhumer.com/opensource/lzo/). It is made available by free software foundation. The algorithm is tested on various operating systems like UNIX and Red Hat Linux.

LZO implements several algorithms with the following features

- 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 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.

5.2 Files required for LZO algorithm

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

5.3 Decompression steps

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

The first field is compressed or decompressed.
The second field is the number of packets 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 CB:        //Broadcast Message
                        {
                               ST_INDEX_DATA*stIndexData = (ST_INDEX_DATA *)cOutputBuf;
                               .
                               .
                               cOutputBuf = cOutputBuf +
                               sizeof(ST_INDEX_DATA); break;
                        }
                }
```

6 Checksum Calculation Algorithm

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

```
// Following is the defines for checksum calculation

#define DC1         17
#define DC3         19
#define CR          13
#define LF          10
#define POLY      0x1021

// End of defines
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);
}
```

7 Glossary

Market Statistics Data
The market statistics data is sent around 18:40 hrs. This data contains the closing details of the government securities / issues / contacts that are trading on that trading day.

End of the Feed
This is an ASCII text sent to intimate the end of the EOD data feed.

Code
This is a unique field in the packet header that describes the type of packet.

Length
Length is a hexadecimal value that contains the length of the packet.

Sequence Number
Sequence Number is an integer that defines the packet number that has been sent by the WDM Infofeed application. Hence the first packet of the day has sequence number 1. It is sequentially incremented by one for subsequent packet.

ASCII Data
It is the data part of the packet.

Trailer
It is a concluding component of each packet.

Checksum
It is a value derived from packet that can be used to check the integrity of the data after transmission. NSE Data & Analytics provides the checksum algorithm.

Carriage Return
Carriage Return

Security Type
The instruments issued by various issuers are clubbed under different homogeneous categories, which are known as Security Types. Security type is a two-character indicator of security depending on the issuer like Central government (GS), state government (SG), public sector unit, institutions, banks, corporates, mutual funds & local bodies.

Security Name
Security name is a description of the security containing, short name of issuer and the year of maturity.

Issue Name
Issue indicates maturity date, coupon rate or mark-up rate over benchmark depending upon nature of the instrument. It is either the rate of interest (in case of coupon bearing instruments) or the date of maturity (in case of non- coupon bearing instruments).

Trade Type
It is a two-character denomination used for WDM trades type. The WDM trading system offers Non-Repo and Repo trades for trading.

Non-Repo Trades
These are trades in which there is an outright purchase and sale of securities. These are denominated by “NR” in the broadcast.

Repo Trades
These trades are repurchased agreements in which a trader sells securities to a customer while simultaneously agreeing to repurchase them at a future date. A Repo transaction involves two phases of trading called “Ready Leg” and “Forward Leg”. In the ready leg phase, the trade is settled as in a normal transaction. In the forward leg phase, the reverse of the trade is settled after the end of the Repo term. These are denominated by “RE” in the broadcast.

Repo Term
It indicates period after which Repo transaction matures.

Number of Trades
It indicates total transactions, in particular security.

Trade High Price
Highest price at which trade takes place in security.

Trade Low Price
Lowest price at which trade takes place in security.

Last Traded Price
It is the price at which the last trade for a particular security has taken place.

Total Traded Value
It is total value of all the securities that have been traded.

Weighted Yield
Weighted yield indicated for the traded security is calculated based on all trades done in the security during the day. It is Yield of each security X Value of each security/ total value of the security.

Market Open
This packet is sent once for each market in day at 10:00 AM. This packet is sent to communicate that the Normal Market is open for trading for that trading day.

Market Close

This packet is sent twice a day at 03:00 pm and 05:45 pm on normal trading days. This packet is sent to communicate market that is closed for trading for that trading day.

Market Type	Normal Trading Days
	Market Close Time
Same Day Market	15:00 Hrs.
Other Day Market	17:45 Hrs.

Trade Information (New Update)

Trade and Order information packet contains the details of the trades/order that have been traded on that trading day on the Exchange.

Heartbeat

Heartbeat is a packet that is sent at regular intervals throughout the day to communicate that the Market environment is up.

Settlement Days

It is the number of days after which the trade will be settled.

Security Status

It is the status of the security available for trading. This can take the following values–

Open - Blank
Participation - ‘P

8 Annexure

8.1 Acronyms Used

BOD	Begin Of Day Information
EOD	End Of Day Information
ONLINE	Information Sent During Market Timing
CM	Cash Market
FO	Future & Options Market
CD	Currency Derivatives Market
WDM	Wholesale & Debt Market
COM	Commodity Market
CBRICS	Corporate Bond Reporting and Integrated Clearing System
NSE IX	NSE International Exchange
MTBT	Multicast Tick By Tick

9 FAQs

1) For Sequenced Data Messages, why do fields contain datatypes as short, but contain value is specified as character?

   Data sent by server contains number, which is the ASCII value of the field and at client’s end it needs to be converted from ASCII value into character.

2) How do you differentiate between numeric and non-numeric values?

   Numeric values are always right aligned and non-numeric values are left aligned. For instance, even though LTP has a datatype as character, it is distinguished by the alignment as numeric value is always right aligned.

3) Can we use lzo versions 2.03/2.09/2.10 for decompressing the packets received from NDAL?

   Yes, lzo is backward compatible. Above versions of lzo can be used for decompressing the compressed packets disseminated from NDAL.

10 Support Information

Name	Email	Contact Number
Business & Technical Support	marketdata@nse.co.in	+91-22-26598385