MARKET FEED
Commodity Market (COM)
Real Time Snapshot Data

Version:1.2

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 September 2018
Version 1.1	1. Added Decoding Snapshot files & FAQs section 2. Data type change for Date field in Contract Information from long to unsigned long	06 October 2025
Version 1.2	Change in the scale and precision for price fields from (6,4) to (7,2)	27 October 2025

Table of Contents

COMMODITY MARKET - SNAPSHOT DATA

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 2 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

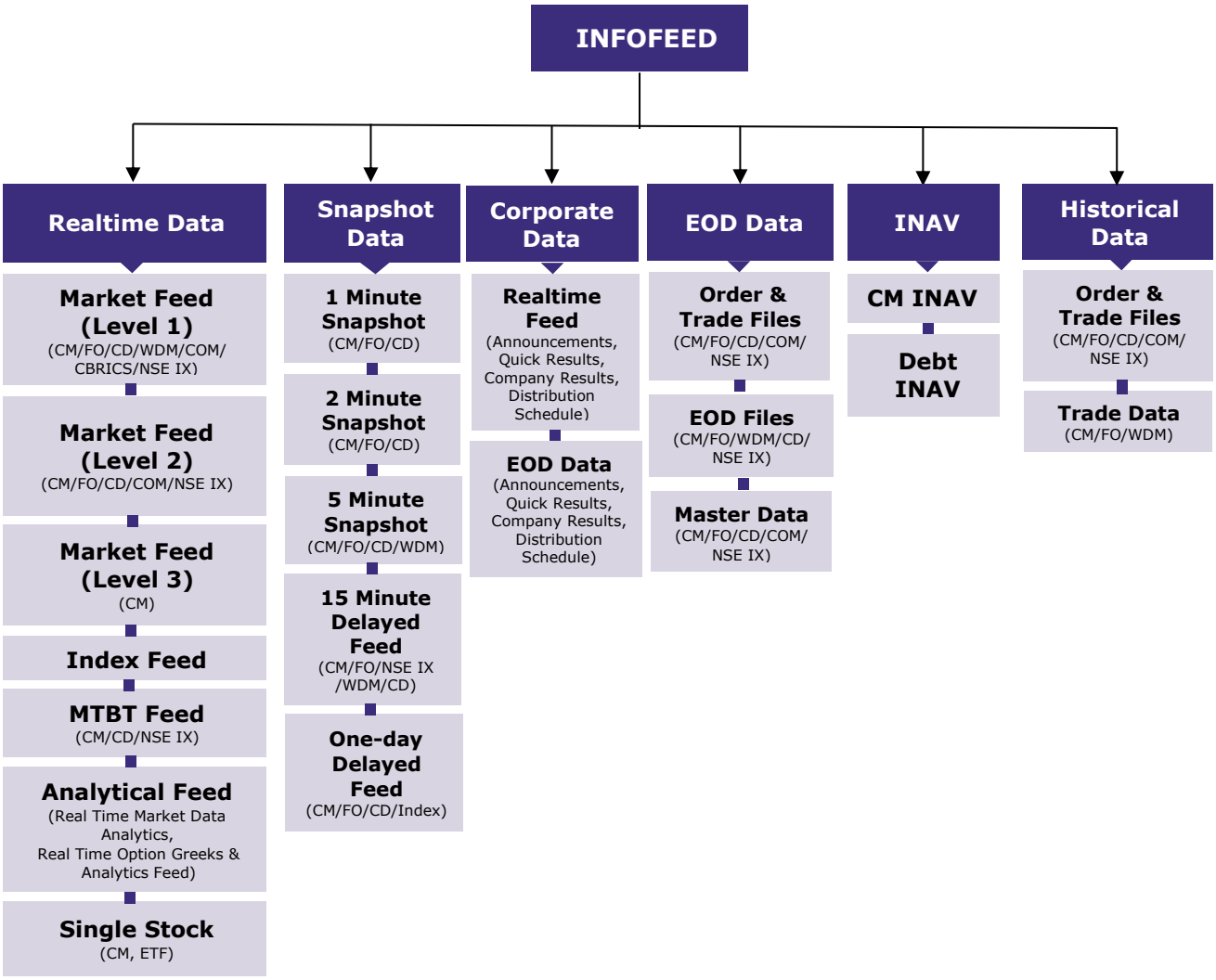
1. End of Day Data
2. Historical Data

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.



This document explains about the NSE – Commodity Market – Snapshot Data. Through this product, binary snapshot files of Commodity data are generated on the server at regular intervals.

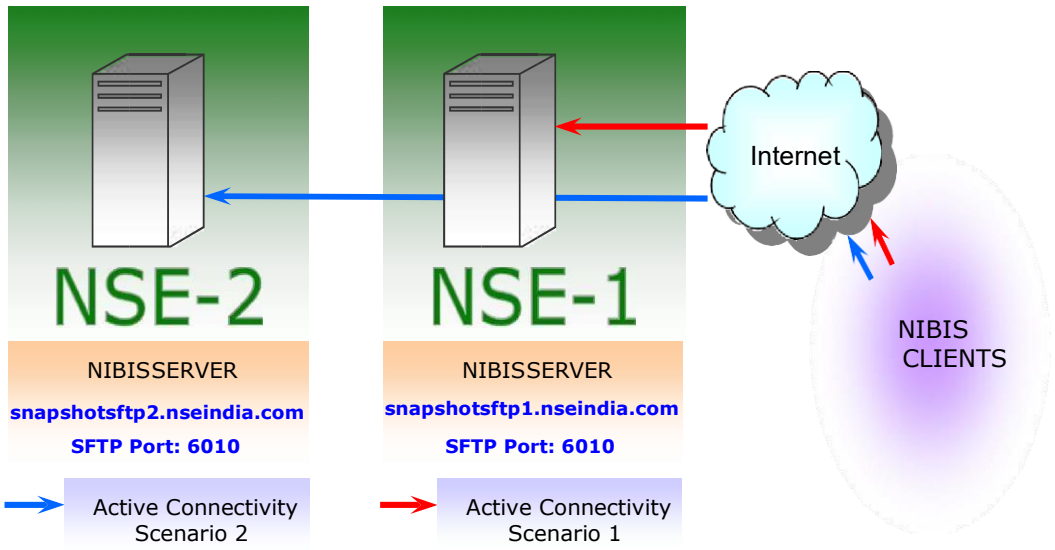
Clients who subscribe to this service connect to the NIBIS (NSE Internet-Based Information System) via the internet and must use the SFTP protocol to download the files. These files are provided in binary format.

To gain access to the NIBIS server, clients are required to submit their server's public key and static public IP address. Once verified, credentials are issued and remain valid for the duration specified in the service agreement.

2 Connection Details

Clients establish a connection to the NIBIS server over the Internet by utilizing the SFTP protocol through a designated public static IP address. Two NIBIS Production Servers operate in an active-active configuration. Each server can be accessed using two IP addresses, for ISP-level redundancy, as shown in the Structural Diagram. In case a server becomes inaccessible through both its IP addresses, the clients are required to fail-over to the other server.

2.1 Structural diagram



2.2 Platform notes

1. The SFTP service can be simultaneously accessed through both redundant IP addresses on each server. This is to enable clients to access the servers in case of link failure.
2. There may be slight differences between the data disseminated by the two servers because of factors impacting sampling such as CPU clock skew, differences in routing of data, etc.
3. Time stamp on the files on the server is in 24-hour format.
4. Certain files are compressed using ZLIB (gzip). The files may be decompressed using the popular "gunzip" command on Unix/Linux systems. Tools to decompress these files are also freely available for Windows on the World Wide Web, Gzip for Windows and 7-Zip being popular examples.
5. The Exchange does not provide software or support for decompression, SFTP, etc.

3 Overview

3.1 Products and "Product Root"

The files are productized as per the generation frequency and are generated under their designated Product Roots on the server.

Product Root is the name of the top-level directory under which files for a product are generated.

Snapshot Frequency	Generation Frequency	Product Root
1 Minute Snapshot Product	Every 1 Minute	/COM01
5 Minute Snapshot Product	Every 5 Minutes	/COM05

The Product Roots may further contain subdirectories as specified in the relevant sections of this document.

Clients may subscribe to product of their choice depending on their data snapshot frequency requirement.

3.2 Types of files generated

The files are generated in binary format on the servers inside the corresponding type-wise sub-directories as specified in this document and can be broadly classified as follows:

Description	Frequency
Market Information Files	At a specific interval
Contract Information Files	Once a day (EOD)
Bhavcopy Information Files	Once a day (EOD)

For each trading day, files are generated in date-wise sub-directories prefixed with the full month name (MonthDDYYYY) as specified in the relevant sections of this document.

The files generated at fixed frequency are continuously numbered, starting from 1.

3.3 Compression

Certain files are compressed using ZLIB (gzip). The files may be decompressed using the popular "gunzip" command on Unix/Linux systems. Tools to decompress these files are also freely available for Windows on the World Wide Web, Gzip for Windows and 7-Zip being popular examples. It may be noted that the Exchange does not provide software or support for decompression.

3.4 Data Types

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

Byte order - Little Endian
All structures are pragma pack 1.

4 Data Details

4.1 Market Information

The Market information data files (MBP and OI) are generated on the server at regular intervals.

4.1.1 Market Files

The *.mkt ("*" stands for a number) files contain market statistics and order information of the contracts that are being traded during the last interval, including their open, high, low and close price. The file contains a single record for every contract that is traded during that file interval. These files are generated during normal trading period To Close Session These files are generated in incremental count number on a trading day starting from 1 (for example, 1.mkt, 2.mkt and so on).

During the special trading session (mahurat trading), the first file will be generated after 15 minutes after the trading session starts.
Actual timing circular will be published by the exchange at the appropriate time.

4.1.2 OI Files

The *.oi ("*" stands for a number) files contain the details of open interest information of contracts. These files are generated at regular intervals. These files are generated during normal trading period To Close Session. These files are generated in incremental count number on a trading day starting from 1 (for example, 1.oi, 2.oi and so on).

4.2 Contract Information

The Contract.dat file is the master file that contains the updated information of all contracts traded on the Exchange. The clients need to download this file and decode it to resolve the "token number" of required contract. The Token number of each contract is unique.

4.3 Bhavcopy Information

The bhavcopy information file is generated after trading hours. The file name is COMBhavcopy_DDMMYYYY.txt. This file contains the End of the Day values of the contracts that are traded on that trading day.

Note: Dummy symbols shall be received in MBP file in an affixed pattern where the last 7 character shall remain "**NSETEST**". Kindly ignore dummy contracts trade volume and pricing information which is coming in MBP file.

5. Data Structure Details

5.1 Market Information

Directory Path	/<Product Root>/DATA/<MonthDDYYYY>
File Name	*.mkt
Compression	Compressed (.gz)
Generation Frequency	At fixed intervals

Field Name	Data Type	Value	Brief Description
INFO HEADER			
Timestamp	LONG	Numeric	This field should be set to numeric zero while sending to the host. This is used in host end.
Message Length	SHORT	Numeric	This field should be set to the length of the entire message, including the length of message header while sending to host.
Info Header Length		**6 Bytes**	
INFO DATA			
Contract Token	LONG	Numeric	This field contains the token number
Market Type	CHAR	Character	N-Normal Market
Best Buy Price	CHAR [17]	Character	The highest price for a Buy auction. Precision up to 2 decimal places.
Best Buy Quantity	CHAR [12]	Character	This field contains the total Buy Quantity
Best Sell Price	CHAR [17]	Character	This is the lowest price for a Sell auction. Precision up to 2 decimal places.
Best Sell Quantity	CHAR [12]	Character	This field contains the total Sell quantity
Last Traded Price	CHAR [17]	Character	This field contains the price at which the latest trade in a

			contract has taken place. Precision up to 2 decimal places.
Total Traded Quantity	CHAR [12]	Character	This field contains the total quantity of a contract traded on the current day
Average Traded Price	CHAR [17]	Character	This field contains the average price of all the trades in a contract. Precision up to 2 decimal places.
Contract Status	CHAR	Character	Blank for active and "S" for suspended
Open Price	CHAR [17]	Character	This field contains the open price of a contract. Precision up to 2 decimal places.
High Price	CHAR [17]	Character	This field contains the highest trade price. Precision up to 2 decimal places.
Low Price	CHAR [17]	Character	This field contains the lowest trade price. Precision up to 2 decimal places.
Close Price	CHAR [17]	Character	This field contains the closing price of a contract. Precision up to 2 decimal places.
Interval High Price	CHAR [17]	Character	This field contains the highest trade price during interval. Precision up to 2 decimal places.
Interval Low Price	CHAR [17]	Character	This field contains the lowest trade price during interval. Precision up to 2 decimal places.
Interval Open Price	CHAR [17]	Character	This field contains the open price of a contract during interval. Precision up to 2 decimal places.
Interval Close Price	CHAR [17]	Character	This field contains the closing price of a contract during interval. Precision up to 2 decimal places.
Interval Total Traded Quantity	CHAR [12]	Character	This field contains the total quantity of a contract traded during interval

Info Data Length	258 Bytes	

5.2 Open Interest Information

Directory Path	/<Product Root>/DATA/<MonthDDYYYY>
File Name	*.oi
Compression	Compressed (.gz)
Generation Frequency	At fixed intervals

Field Name	Data Type	Value	Brief Description
INFO HEADER			
Timestamp	LONG	Numeric	This field should be set to numeric zero while sending to the host. This is used in host end.
Message Length	SHORT	Numeric	This field should be set to the length of the entire message, including the length of message header while sending to host.
Info Header Length		**6 Bytes**	
INFO DATA			
Contract Token	LONG	Numeric	Token number of the index being updated.
Market Type	CHAR	Character	This field contains the opening index value at the time of market open.
Open Interest	CHAR [12]	Character	This field contains the online market index value at that instance of broadcast.
Info Data Length		**17 Bytes**	

5.3 Contract Information

Directory Path	/<Product Root>/CONTRACT/<MonthDDYYYY>
File Name	Contract.dat
Compression	Not compressed
Generation Frequency	Once (EOD)

Field Name	Data Type	Value	Brief Description
INFO DATA			
Token Number	LONG	Numeric	Token number of the contract being updated.
Instrument Name	CHAR [7]	Character	This field contains the type of the instrument.
Symbol	CHAR [11]	Character	This field contains the Symbol of the contract.
Series	CHAR [3]	Character	This field contains the series of a contract.
Expiry Date	UNSIGNED LONG	Numeric	This field contains the last date of trading before any corporate action. Expiry Date is sent in epoch format.
Strike Price	LONG	Numeric	This field will contain a valid strike for Options Contract and for Futures Contract it will be -1. The client should divide the strike price by 100.
Option Type	CHAR [3]	Character	Option Type
Issue Start Date	UNSIGNED LONG	Numeric	Date of issue of the contract.
Issue Maturity Date	UNSIGNED LONG	Numeric	Maturity date
Board Lot Quantity	LONG	Numeric	Regular lot size.
Tick Size	LONG	Numeric	Tick size/ Min spread size
Contract Name	CHAR [26]	Character	This field contains the contract name.

Record Date	UNSIGNED LONG	Numeric	Date of record changed.
Ex Date	UNSIGNED LONG	Numeric	Last date of trading before any corporate action.
No Delivery Start Date	UNSIGNED LONG	Numeric	Date from when physical delivery of share certificates is stopped for book closure
No Delivery End Date	UNSIGNED LONG	Numeric	No delivery end date.
Book Closure Start Date	UNSIGNED LONG	Numeric	Date at which the record books in the company for shareholder names start.
Book Closure End Date	UNSIGNED LONG	Numeric	Date at which the record books in the company for shareholder names end
Remarks	CHAR [26]	Character	This field contains the remarks
Info Data Length		**128 Bytes**	

5.4 Bhavcopy Information

This data file does not contain the Header field.

Directory Path	/<Product Root>/BHAVCOPY/<MonthDDYYYY>
File Name	COMMKTSTATSDDMMYYYY.DAT
Compression	Not compressed
Generation Frequency	Once (EOD)

Field Name	Data Type	Value	Brief Description
INFO DATA			
Instrument Name	CHAR [6]	Character	Instrument Name
Symbol	CHAR [10]	Character	This field contains the Symbol of the contract.
Expiry Date	CHAR [10]	Character	Expiry Date (DD-MM-YYYY)
Strike Price	CHAR [10]	Character	Precision up to 2 decimal places.
Option Type	CHAR [2]	Character	Option Type. Valid values are: CE -- CALL OPTION PE -- PUT OPTION FF -- FUTURES CONTRACT
Market Type	CHAR	Character	N - Normal Market
Opening Price	CHAR [17]	Character	This field contains the indicative opening price (IOP) of a contract for order collection period session and Final Open Price of a contract in matching period. Precision up to 2 decimal places.
Trade High Price	CHAR [17]	Character	This field contains the highest trade price. Precision up to 2 decimal places.
Trade Low Price	CHAR [17]	Character	This field contains the lowest trade price. Precision up to 2 decimal places.
Closing Price	CHAR [17]	Character	This field contains the closing price of a contract. Precision up to 2 decimal places.

Last traded Price	CHAR [17]	Character	Precision up to 2 decimal places.
Previous Close Price	CHAR [17]	Character	This field contains the previous day's closing price of the contract. Precision up to 2 decimal places.
Settlement Price	CHAR [17]	Character	Precision up to 2 decimal places.
Total Traded Quantity	CHAR [12]	Character	This field contains the total quantity of a contract traded on the current day
Total Traded Value	CHAR [25]	Character	This field contains the total value of the contracts traded
Current Open Interest	CHAR [10]	Character	This field contains the Current Open Interest of the contract.
Change in Open Interest	CHAR [10]	Character	This field contains the change in value of open interest.
Info Data Length		**215 Bytes**	

6. Notes

6.1 All prices are in Paisa.

All price fields are multiplied by 100 and this implies that the prices received must be divided by 100.

6.2 Timestamp

The timestamp is the number of seconds elapsed from midnight Jan 1, 1980. All the dates we are sending in epoch format.

7. About SFTP (Secure File Transfer Protocol)

The file transfer takes place over SFTP (Secure FTP) protocol over the Internet.

The client is required to submit the SSH RSA Public Key of their machine along with their static public IP address to receive access details from NSE Data & Analytics (NDAL).

The following details will be provided once the request is processed by NDAL:

- URL
- SSH Service Port
- User ID
- File Path

General information on SFTP has been provided in the following sections for popular OS platforms.

7.1 SFTP on Linux platform

The OpenSSH suite, which comes pre-installed in most Linux distributions, can be used for transferring files securely using SFTP.

The SSH key-pair is generally generated in the ".ssh" directory in the user's home directory.

It is highly recommended that you consult your systems administrator to generate/locate the key-pair and set up SFTP for you.

Continue reading for information on how to generate the key-pair.

7.1.1 Generation of the SSH RSA key-pair on Linux

- Generate the new key-pair with following command:
  `ssh-keygen -t rsa -C "user@host"`

- You will receive the following prompt:
  `Generating public/private rsa key pair.`
  `"Enter file in which to save the key".`
  Press the Enter to continue with the defaults.

  You will receive the following prompt:
  `Enter file in which to save the key.`
  `(/host/users/user/.ssh/id_rsa):`
  Press the Enter to continue with the defaults.

- If a file already exists with the same name, then you will receive the following prompt:
  ```
  /host/users/user/.ssh/id_rsa already exists.
  Overwrite (y/n)?
  ```
  Type "y" and press Enter to overwrite.

- You will be prompted to enter a passphrase as follows:
  ```
  Enter passphrase (empty for no passphrase):
  ```
  Press Enter to continue without a passphrase.

  Your will be prompted to re-enter the passphrase:
  ```
  Enter same passphrase again:
  ```
  Press Enter again to continue without a passphrase.

- After you enter a passphrase, you will be presented with the "Fingerprint" (or ID) of your SSH key.

  It will look something like this:
  ```
  Your identification has been saved in
  /host/users/user/.ssh/id_rsa.
  Your public key has been saved in
  /host/users/user/.ssh/id_rsa.pub.
  The key fingerprint is:
  87:c4:85:90:91:16:39:de:c2:26:49:4a:b3:38:80:97 user@host
  ```

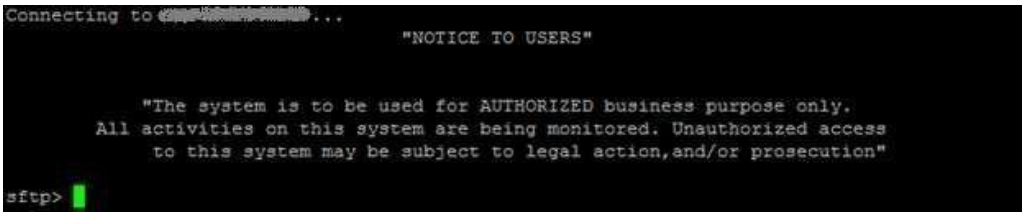
  After generating public key, user needs to share the Public Key file with exchange for requesting the credentials.

NOTE: In above steps the words "host" and "user" are used to represent the host name and username of the machine. This is used for demo purpose only. The same will differ as per your server and usernames.

7.1.2. SFTP Login

Login to the Exchange Server over SFTP using the following command:
```
sftp -o PORT=6010 remote_user@remote_host
```

Where remote user is the User ID provided to you by the Exchange upon sharing your Public Key and remote host is the Exchange Server IP.
You should get the SFTP prompt as below, upon successful login:



7.1.3 Fetching files over SFTP

The SFTP "get" command may be used at the SFTP prompt for fetching the files while logged into the host over SFTP.

7.1.4 Ending the SFTP session.

The SFTP "bye" command may be used for terminating the session.

7.1.5 SFTP commands help.

Help may be obtained with SFTP commands by typing the "help" command at the SFTP prompt.

7.2 SFTP on Windows platform

7.2.1 Generation of the SSH RSA key-pair on Windows

This guide explains how to generate the SSH RSA key-pair using the PuttyGen application.

Download the PuttyGen application (freely available on the Internet). Then follow these steps to generate the key-pair:

• Start the PuttyGen application.
You will be presented with a dialog which looks something like this:

- Select "SSH2RSA" with 2048-bit size or greater.
- Press the "Generate" button.
- After generating the key, you will be shown the screen below. Keep the "Key passphrase" and "Confirm passphrase" as blank.

SSH RSA Public Key should be copied from here after generation

- Create a blank file with the name "id_rsa.pub".
  This will be the public key file which will be populated with your Public Key and shared with the Exchange.

- Copy the public key content as presented on the screen (selected area in the below screenshot) and paste into newly created public key file (id_rsa.pub) and save the file.

- Share this Public Key File (id_rsa.pub) with the Exchange when requesting for SFTP credentials.

7.2.2 SFTP Client Software on Windows

There are multiple SFTP Client Programs (paid for and free) available for transferring files over SFTP.

One such software is WinSCP, available for free from the WinSCP website.
This program is intuitive, user friendly and can be used in interactive mode (GUI) as well as from the command line (for automation/batch processing).

Information on using WinSCP can be found on the WinSCP website.

7.3 Further support

Apart from the above guide, many of the online resources can be referred on the World Wide Web for more information on how to set up and use SFTP at the Client's site on various OS platforms.

Note: This "About SFTP" section is intended as a guide used to understand and become familiarized with this transfer protocol.

It may be noted that the Exchange does not provide SFTP software or support for configuring and using SFTP at Client site.

8. Decoding Snapshot files

Please refer following C snippet for decoding snapshot files:

```
HEADER header     = {0};
DATA_MBP data_mbp = {0};
DATA_OI data_oi = {0};
TRAILER trailer   = {0};

int main()
{
 gzFile  fpFile = gzopen("1.mkt.gz", "rb");
 while
 {
      int nRetVal = gzread(fpFile, &header ,sizeof(HEADER));
      if(header.nTranscode == MBP)
      {
            gzread(fpFile, &data_mbp, sizeof(data_mbp));
            print(&data_mbp);
      }

      else if(header.nTranscode == OI)

      { gzread(fpFile, &data_oi, sizeof(data_oi)); }

      nRetVal = gzread(fpFile, &trailer, sizeof(TRAILER));
      if(gzeof(fpFile) == 1)
      {
                  printf("EOF reached ");
                  break;
      }
 }
 gzclose(fpFile);
}

void print(DATA_MBP *data_mbp)
{
 printf("Contract Token = %d ,Last Traded Price = %d,
  Best Buy Quantity = %d,  ....." ,data_mbp->ContractToken,
  data_mbp->LastTradedPrice,data_mbp->BestBuyQuantity);
}
```

9. FAQs

1) Download of files through SFTP was working till last week, suddenly our connection to sftp is failing. How do we resolve it?

   If using SFTP on Windows, please ensure you are using the latest version of Winscp or any other equivalent tool.

   If you are using SFTP programmatically or through an API, please ensure you **don't use the following cipher**:

   - diffie-hellman-group-exchange-sha1
   - diffie-hellman-group14-sha1
   - diffie-hellman-group1-sha1

2) How do we decode compressed *.mkt and *.oi files?

   Please refer to section 8 for decoding Snapshot Files, which involves decompression and reading of *.mkt and *.oi files.

10. Contact Information

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