MARKET FEED
Wholesale Debt Market (WDM)
(15 Mins Delayed Snapshot)

Version: 1.0
Date: 24 June 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 June 2025

Table of Contents

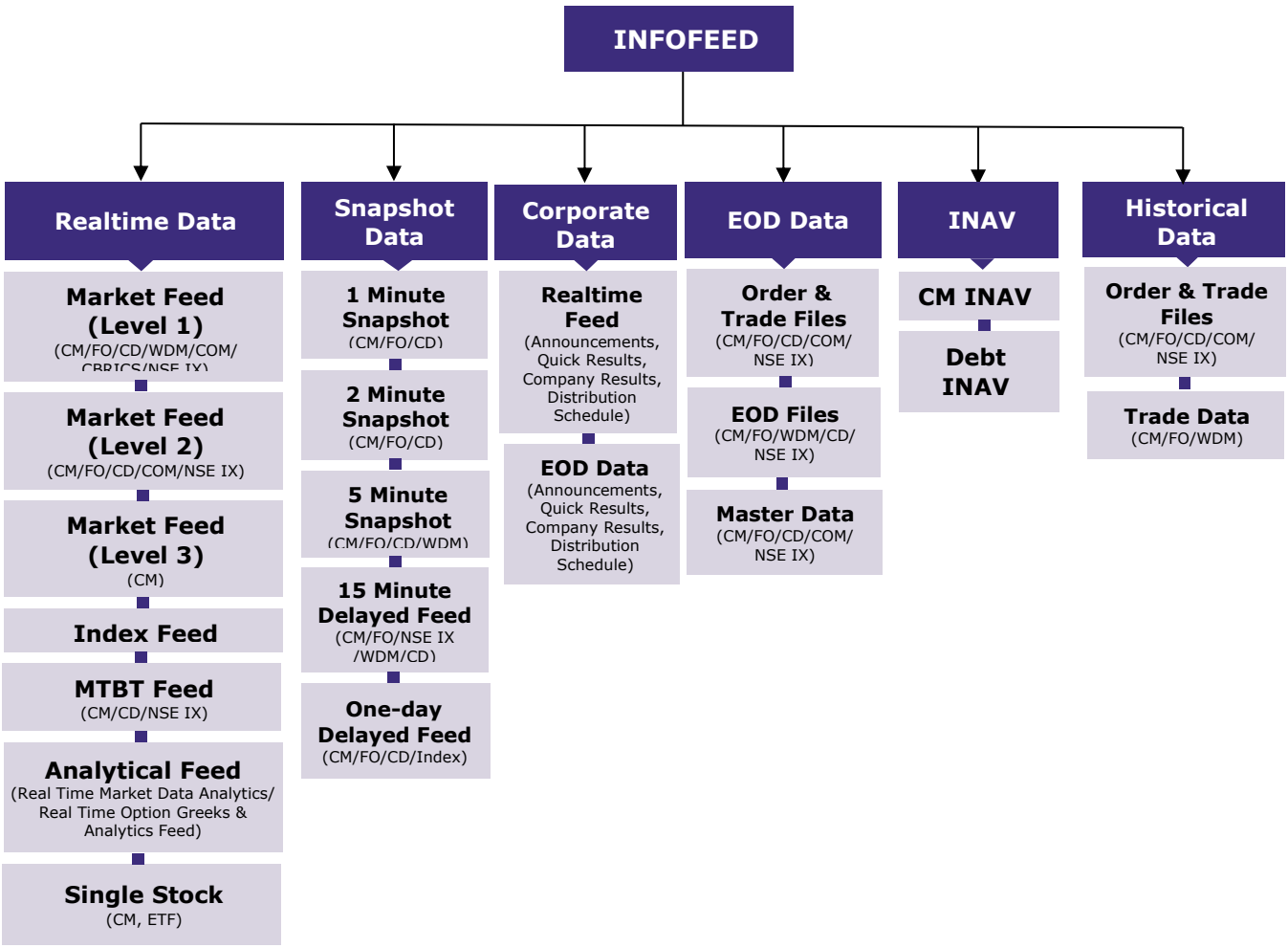
Market Feed – Wholesale Debt Market (15 Minutes Snapshot)

1 Introduction

NSE Data & Analytics Ltd. disseminates NSEIL's suite of data products to various information agencies. It provides 7 different types of data products viz.

1. Real Time Data
2. Snapshot Data
3. End of Day Data
4. Corporate Data
5. Analytical Products data
6. Historical Data
7. Indicative NAV Data

The Realtime data and corporate data is a packet broadcast available for dissemination through feed, whereas the snapshot data, end of day data and historical data is available in the form of files. All these data products come under Infofeed application.



The NIBIS (NSEIL's Internet Based Information System) server that caters the NIBIS clients is available through internet. All NIBIS clients are required to connect the server through internet and use SFTP protocol to download the files.

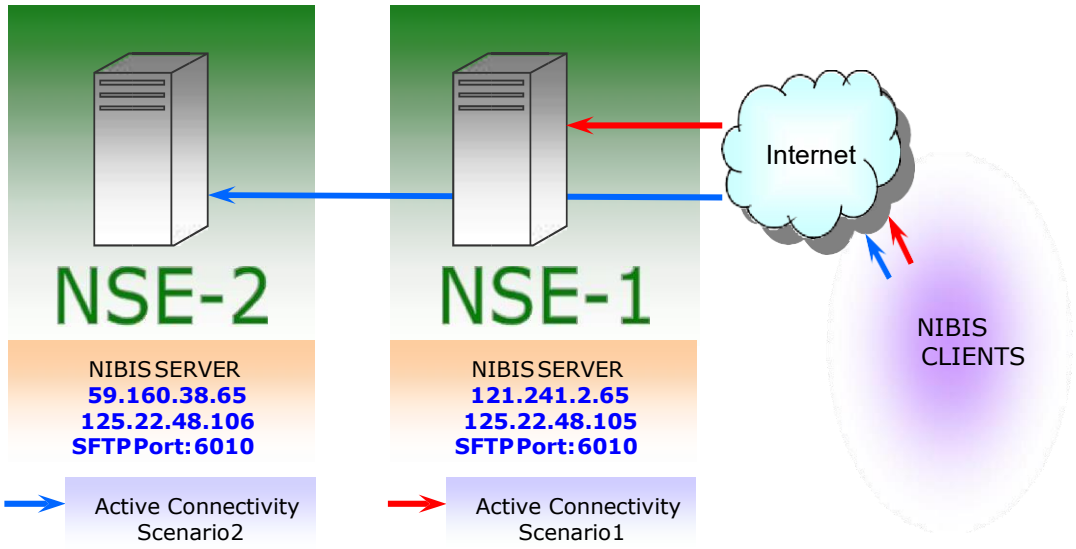
The files on this server are generated with 15 minutes delay on regular 1 minute interval basis. These files are generated in binary format on the server and at regular intervals.

To gain access to the NIBIS server, clients must provide their server public key and public static IP address. Credentials are issued to clients and are valid for the duration specified in the agreement.

2. Connection Details

The Info Clients connect to NIBIS server over the Internet using SFTP protocol. In NSE premises 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 Info Client software requires to fail-over to the other server.

2.1. Structural Diagram



2.2 Data Types

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

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

2.3 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. Data Details

The Wholesale Debt Market (WDM) data files are generated in binary format at a regular interval of 1 minute (15 minutes old data).

3.1 Market Information

The Market information data files are generated in /WDM30/DATA/" <Month>DDYYYY" folder on the server. i.e., *.mkt

The *.mkt (Where "*" stands for a numeric value) files contains market statistics and order information of the securities that are being traded 15 minutes before and generated with 1 minute interval along with their open, high, low and close price. The file contains a single record for every security that is traded during that file interval. These files are generated in incremental count number on a trading day starting from 1.mkt.

During 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. Data Structure Details

4.1 Market Information

FILE PATH - /WDM30/DATA/" <MONTH>DDYYYY" FILE NAME - *.MKT

Field Name	Data Type	Value	Brief Description
FILE HEADER			
Timestamp	LONG	Numeric	File generation time
Number of Records	LONG	Numeric	Number of records in the file
File Header Length		**8 Bytes**	
RECORD HEADER			
Transcode	SHORT	Numeric	Transaction message number. This describes the type of packet received or sent.
Timestamp	LONG	Numeric	Time when the record is updated
Message Length	SHORT	Numeric	Size of DATA packet
Record Header Length		**8 Bytes**	
RECORD DATA			
Security Type	CHAR [2]	Character	Security Type of Bond
Security Name	CHAR [7]	Character	Security Name of Bond
Issue Name	CHAR [6]	Character	Issue Name of Bond
Settlement Days	CHAR [3]	Character	Settlement Period
Trade Type	CHAR [2]	Character	Security Trade Type
Repo Term	CHAR [3]	Character	Repo Term
Trade High Price	CHAR [10]	Character	High Price
Trade Low Price	CHAR [10]	Character	Low Price
Last Traded Price	CHAR [10]	Character	Last Traded Price
Total Traded Value	CHAR [15]	Character	Total Traded Value
Security Status	CHAR [1]	Character	Security Status
Checksum	CHAR [12]	Character	Checksum
Info Data Length		**81 Bytes**	

5. File Transcode List

DETAILS	TRANSCODE NUMBER
MARKET OPEN INFORMATION	0x4E57

6 Data Field Details

6.1 Transcode

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

6.2 Timestamp

The timestamp is the number of seconds elapsed from midnight Jan 1, 1980.

6.3 Message Length

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

6.4 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 the security depending on the issuer like Central government (GS), state government (SG), public sector unit, institutions, banks, corporate, mutual funds & local bodies.

6.5 Security Name

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

6.6 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).

6.7 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.

6.7.1 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.

6.7.2 Repo Trades

These trades are repurchasing 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.

6.8 Repo Term

It indicates period after which Repo transaction matures.

6.9 Number of Trades

It indicates total transactions in a particular security.

6.10 Trade High Price

Highest price at which trade taken place in a security.

6.11 Trade Low Price

Lowest price at which trade taken place in a security.

6.12 Last Traded Price

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

6.13 Total Traded Value

It is total face value of all the securities that have been traded.

6.14 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.

6.15 Market Open

WDM Market opens on each trading day at 10:00 Hrs.

6.16 Market Close

WDM Market closes on each trading day at following timings –

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

6.17 Settlement Days

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

6.18 Security Status

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

Status	Value
Open	Blank
Suspended	S

7. Date and Time Conversion

7.1 Sample Program 1

Sample program for converting long date into the DD MM YYYY format.

```
/*********************************************************
Routine Name: DateConv
Synopsis: This routine is responsible for processing a
Date (LONG) which constitutes a six-digit integer.
The integer is then converted into the following:
A string DD MMM YYYY
If the incoming number is zero it will return blank date.
Parameter descriptions:
lNoInput - Incoming number
pDateStr - Date output in string format
Return value: NONE
*********************************************************/

#define JAN 0
#define FEB 1
#define MAR 2
#define APR 3
#define MAY 4
#define JUN 5
#define JUL 6
#define AUG 7
#define SEP 8
#define OCT 9
#define NOV 10
#define DEC 11
#define JAN_STR "JAN"
#define FEB_STR "FEB"
#define MAR_STR "MAR"
#define APR_STR "APR"
#define MAY_STR "MAY"
#define JUN_STR "JUN"
#define JUL_STR "JUL"
#define AUG_STR "AUG"
#define SEP_STR "SEP"
#define OCT_STR "OCT"
#define NOV_STR "NOV"
#define DEC_STR "DEC"

CHAR m_cMonth[12][4];
strcpy(m_cMonth[JAN],JAN_STR);
strcpy(m_cMonth[FEB],FEB_STR);
strcpy(m_cMonth[MAR],MAR_STR);
strcpy(m_cMonth[APR],APR_STR);
strcpy(m_cMonth[MAY],MAY_STR);
strcpy(m_cMonth[JUN],JUN_STR);
strcpy(m_cMonth[JUL],JUL_STR);
strcpy(m_cMonth[AUG],AUG_STR);
strcpy(m_cMonth[SEP],SEP_STR);
```

```
strcpy(m_cMonth[OCT],OCT_STR);
strcpy(m_cMonth[NOV],NOV_STR);
strcpy(m_cMonth[DEC],DEC_STR);

BOOL DateConv (long lNoInput ,char * pDateStr )
{
   struct tm *pDate ;
   static char cConvertedMonth [ 4 ];
   char s1[]="19";
   char s2[]="20";
   int j;
   CString cTempMonth;
   cTempMonth = ' ';
   if ( lNoInput == 0L )
   {
      strcpy(pDateStr,"");
   }
   else
   {
       // Convert the incoming number...
       lNoInput += 315513000L ;
       pDate = localtime ( ( time_t * ) &lNoInput ) ;
       if(pDate == NULL)
       {
          return FALSE;
       }
       if(pDate->tm_mon>=0 && pDate -> tm_mon<12)
       {
          strcpy(cConvertedMonth,m_cMonth[pDate->tm_mon]);
       }

       //NOW THE STRING FORMAT
       j=sprintf(pDateStr,"%02d%03s", pDate -> tm_mday , cConvertedMonth);
       if (pDate->tm_year > 99)
       {
          // if year after two thousand bring year back to two
          digits
          pDate->tm_year -= 100 ;
          sprintf ( pDateStr +j,"%02s%02d",s2, pDate -> tm_year);
       }
       else
       {
          sprintf ( pDateStr +j,"%02s%02d",s1, pDate -> tm_year);
       }
   }
   return TRUE;
}
```

7.2 Sample Program 2

Sample program for convert long time into HH:MM:SS format

```
/*************************************************************
Routine Name: TimeConv
Synopsis: This routine is responsible for processing a time
Value (LONG) which constitutes a six-digit integer.
The integer is then converted into the following:
(takes into account decade 70 )
A string - 99:99:99
If the incoming number is zero it will return blank time
Parameter descriptions:
   lNoInput - Incoming number
   cTimeStr - Time output in string format
Return value: NONE
*************************************************************/
VOID TimeConv(LONG lNoInput,CHAR* cTimeStr)
{
   struct tm * pTimeStruct ;
   CString szTime;
   if ( lNoInput == 0 )
   {
      strcpy(cTimeStr,"");
   }
   else
   {
      pTimeStruct = localtime ( ( time_t * ) &lNoInput ) ;
      if ( pTimeStruct->tm_isdst == 1 )--pTimeStruct->tm_hour ;
      if (pTimeStruct->tm_hour == -1 )
      {
         // Make sure that from midnight returns to 11 pm.
         pTimeStruct->tm_hour = 23;
      }
      sprintf ( cTimeStr, "%02d:%02d:%02d",
         pTimeStruct->tm_hour, pTimeStruct->tm_min,
         pTimeStruct->tm_sec);
   }
}
```

8. About SFTP (Secure File Transfer Protocol)

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

The Info Client requires to provide the Exchange with the SSH RSA Public Key of their machine for receiving login details form the Exchange.

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

- Server IP
- SSH Service Port
- User ID
- File Path

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

8.1 SFTP on Linux platform

The Open SSH 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.

8.1.1 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.

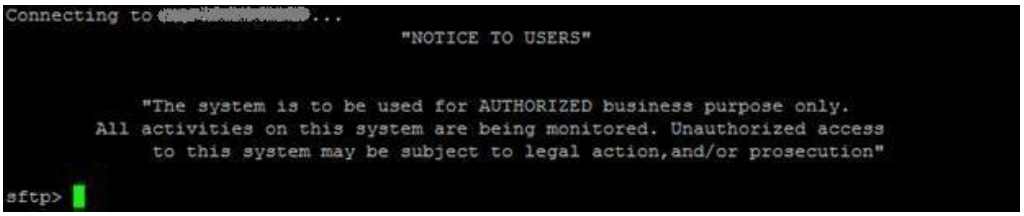
NOTE: In above steps the words "host" and "user" are used to represent the host name and username of the machine. This is used for demo purpose only. The same will differ as per your server and usernames.

8.1.2 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:



8.1.3 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.

8.1.4 Ending the SFTP session

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

8.1.5 SFTP commands help

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

8.2. SFTP on Windows platform

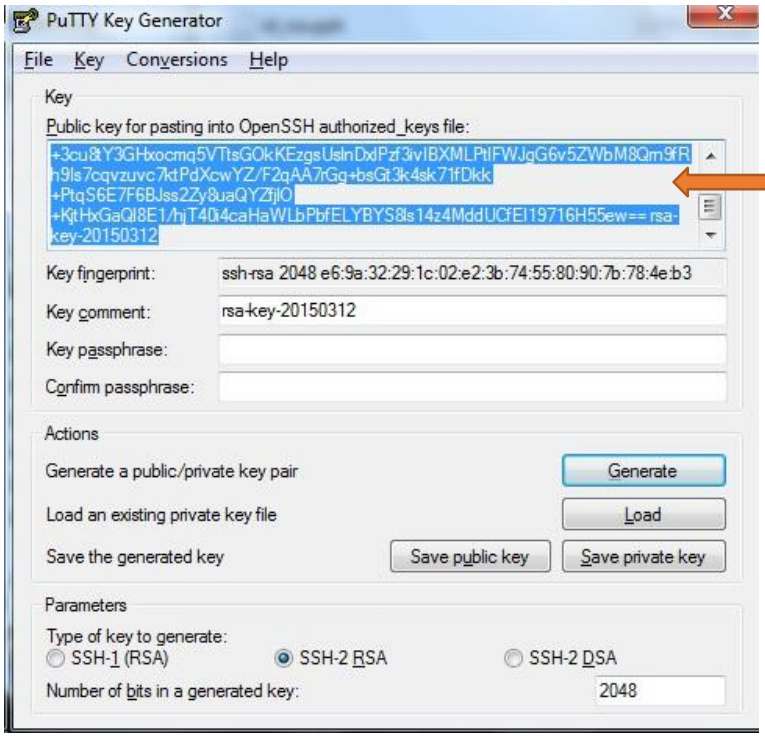
8.2.1. Generation of the SSH RSA key-pair on Windows

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.

8.2.2. 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.

8.3. 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

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

10. Contact Information

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