TECHNICAL DOCUMENT

REAL TIME DATA

WHOLESALE DEBT MARKET
(LEVEL 1)

24 APR 2009

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

COPYRIGHT NOTICE

CONTENTS

REAL TIME DATA TECHNICAL SPECIFICATION

WHOLESALE DEBT MARKET

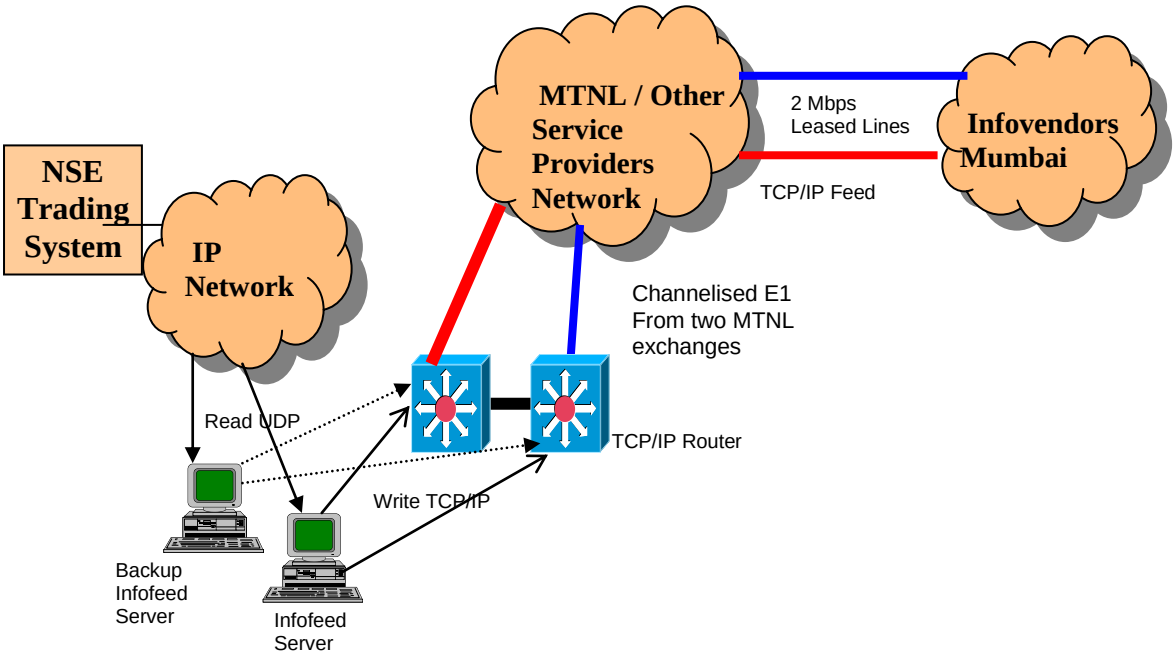
1. INTRODUCTION

DotEx international Ltd. disseminates NSEIL's real time broadcast data to various information agencies. It provides the 3 different types of data to vendors, i.e. Real Time Data, Snapshot Data and End of Day Data. The real time data is a packet broadcast available in TCP/IP format, where as the snapshot data and End of day data is available in the form of files. The Infofeed server provides NSEIL real time broadcast data. The information agencies connect to the Infofeed Server through 2Mbps Leased Lines. These leased lines are terminated on Infofeed Router and their data specific pneumonic calls are forwarded to Infofeed server. The Infofeed server accepts these pneumonic calls and creates a socket connection. The TCP/IP data flows to the information agencies through these socket connections

2. CONNECTION DETAILS

Structural Diagram

The structural diagram of Infofeed connection has been explained below



Online Requirements

a. A Router / Switch or a card with X25 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.

Steps for Authentication AND Receiving feed :

a) Client applications at vendor end, establish the connection with Infofeed Server application using specified IP address and Port.
b) After establishing the connection, client application sends the login packet to Infofeed server application.

Packet format of Login packet(LOGIN_REQ) is as follows,

```
struct LOGIN_REQ
{
        struct INFO_HEADER sHeader;
        struct LOGIN_REQ_DATA sData;
        struct INFO_TRAILER sTrailer;
};
struct LOGIN_REQ_DATA {
        char cUserId[10];
        char cPassword[8];
        char cNewPassword[8];
        char cConfmPassword[8];
};
struct INFO_HEADER
{
        short iCode;
        short iLen;
        LONG lSeqNo;
};
struct INFO_TRAILER
{
        short iCheckSum;
        CHAR cEOT;
};
```

c) Password field is case sensitive, password should be minimum 6 characters long, password and user id should not be same , password should start with alphabet and password should be alphanumeric (No wild characters are allowed).
d) If user wants to change his password then the user needs to specify new password & confirm password (Both fields should match) otherwise leave it blank.
e) Based on the above information, user will get Log on response (LOGIN_RESPONSE) from Infofeed Server.

```
struct LOGIN_RESPONSE
{
        struct COM_HEADER sCOMHeader;
        struct LOGIN_RESPONSE_DETAIL sDetail;
};
struct LOGIN_RESPONSE_DETAIL
 {
        struct INFO_HEADER sHeader;
        struct LOGIN_RESPONSE_DATA sData;
```

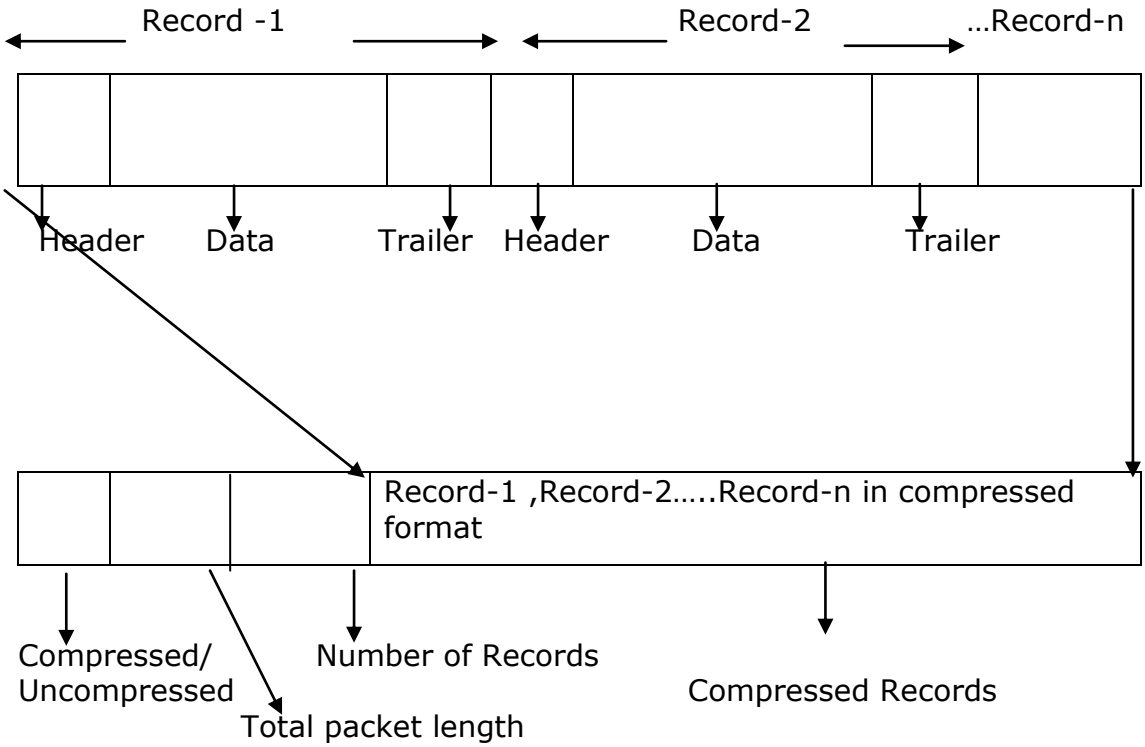
```
        struct INFO_TRAILER sTrailer;
};
struct COM_HEADER
{
        char cCmopOrNot;
        short iPackLen;
        short iNoOfPack;
};
struct LOGIN_RESPONSE_DATA
{
        long iErrCode;
        char cMesg[50];
};
```

Following Error code will be returned which client needs to interpret as:-

1000- Successful
1001- Password Update Successfully
1002- Wrong UserId-Password Combination
1003- Password is not valid in password change request.
1004- Login request header is not correct.
Error code other than above - Error in receiving logon response

f) After successful login, Infofeed Application starts sending packets in the below format.

Packet Format:



Compressed/Uncompressed: This field tells whether packet is compressed or not Compressed.
If this Field = 0 then Compressed.
Field = 1 then Uncompressed.

Number of Records: This field tells the number of records present in the compressed packets.

Packet length: this field specifies the total packet length.

```
Structure COM_HEADER
{
        char cCmopOrNot;
        short iPackLen;
        short iNoOfPack;
};
```

g) As the data packets are sent in compressed format there is a need to decompress them. The compression algorithm used is LZO.
h) The Decompression algorithm used should be LZO

Steps for Decompressing feed :

- LZO Algorithm Details:

    a) LZO is a data compression library which is suitable for data de-/compression in real-time. This means it favors speed over compression ratio.
    b) 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.
- Compression is pretty fast.
- 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 decompressor 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.

c) Files required for LZO algorithm.

- Include files, source files (src) provided by LZO
- LZO.lib

For more information on LZO library and downloads visit:
http://www.nseindia.com/content/press/prs_whatsnew.htm#2

Specifications for utilizing on-line broadcast information

- **Decompression steps**:

a) Receive the packet in the temporary buffer i.e. array of characters.
b) First field will identify whether the packet is compressed or not.
c) If this field is **0** then Decompress it using LZO algorithm else if **1 don't** decompress it and proceed in normal way as it is being done today.
d) The second field is packet length.
e) The third field contains the number of records in the packet.
f)If compressed use following function of LZO to Decompress.

r = lzo1z_decompress ((unsigned char *) cInputBuf, ipLength, (unsigned char*) cOutputBuf, 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

g) After decompression data will be available in Output Buffer.
h) Map the outputbuf to existing Header structure according to **WN.**
i) Look for Record size in the length field and Code (i.e. WN).
j) Steps to recover data from OutputBuf.

Algorithm:

```
Length_of_Record = Header->length;
Sequence_no = Header->Sequence_num;
For I = 0 to Number of records (obtained in step 4)
Begin
   Bytes_to_seek = Length_of_Record * I
   Seek to number of Bytes_to_seek
   Map (Length_of_Record) of bytes to proper structure
   according to
   iCode (WN) as found in Header part.
   Do the required processing….
    ….

   End
End for Loop.
```

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 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) Heart Beat Signal

TRAILER

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

4. DATA STRUCTURE DETAILS

LOGIN REQUEST (send by client application)

Login Request packet will be send by the client application for login into the Infofeed application. If user wants to change his password he will specify the new password and confirm password field. Password is case sensitive. Format of this packet is as follows.

4.1.1 INFO HEADER:

a) Transaction Code	'**WQ**'
b) Length	Short Integer (2 Bytes)
c) Sequence Number	00

4.1.2 ASCII DATA:

a) User Id	10 Chars
b) Password	8 Chars
c) New Password	8 Chars
d) Confirm Password	8 Chars

4.1.3 INFO TRAILER:

Checksum is not computed. A CR will terminate the block of data.

4.2 LOGON RESPONSE (send by Infofeed application)

Logon response packet will be send by the Infofeed server application after receiving the Login Request packet from the client application.

4.2.1 COM HEADER:

a) Compressed or Not	1 Char
b) Packet Length	Short Integer (2 Bytes)
c) No of Packets	1

4.2.2 INFO HEADER:

a) Transaction Code	'**WR**'
b) Length	Short Integer (2 Bytes)
c) Sequence Number	00

4.2.3 ASCII DATA:

a) Error Code	Long (4 Bytes)
b) Message	50 Chars

4.2.4 INFO TRAILER:

Checksum is not computed. A CR will terminate the block of data.

4.3 MARKET OPEN

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

HEADER:

a) Code	'WO'
b) Length	0x6F
c) Sequence Number	XXXX

ASCII DATA:

1) Message	100 Characters

TRAILER:

Checksum is not computed. A CR will terminate the block of data.

4.4 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 packet is sent twice in a day in WDM segment.

HEADER:

a) Code	'WC'
b) Length	0x6F
c) Sequence Number	XXXX

ASCII DATA:

1) Message	100 Characters

TRAILER:

Checksum is not computed. A CR will terminate the block of data.

4.5 TRADE INFORMATION (NEW UPDATE)

HEADER:

a) Code	'WN'
b) Length	0x50
c) Sequence Number	XXXX

ASCII DATA:

The format of the ASCII data sent is as follows:

FIELD TYPE	**FIELD WIDTH**
1) Security Type	2 characters
2) Security Name	7 characters
3) Issue Name	6 characters

4) Settlement Days	3 characters
5) Trade Type	2 characters
6) Repo Term	3 characters
7) Trade High Price	10 characters
8) Trade Low Price	10 characters
9) Last Traded Price	10 characters
10) Total Traded Value	15 characters
11) Security Status	1 character

TRAILER:
The Trailer is a 2-byte checksum. A CR will terminate the block of data.

4.6. HEARTBEAT DATA

Heart Beat is a packet that is sent at regular intervals through out the day to communicate that the Market environment is up.

HEADER:

a) Code	'WH'
b) Length	0x0B
c) Sequence Number	XXXX

ASCII DATA:
Not associated with any ASCII data.

TRAILER:
Checksum is not computed.

4.7. 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.

HEADER:

a) Code	'WS'
b) Length	0x55
c) Sequence Number	XXXX

ASCII DATA:
The format of the ASCII data sent is as follows:

Field Type	**Field Width**
1) Security Type	2 characters
2) Security Name	7 characters
3) Issue Name	6 characters

4) Trade Type	2 characters
5) No. Of Trades	4 characters
6) Trade Value	15 characters
7) Trade Low Price	10 characters
8) Trade High Price	10 characters
9) Last Traded Price	10 characters
10) Weighted Yield	8 characters

TRAILER:
The Trailer is a 2-byte checksum. A CR will terminate the block of data.

4.8. END OF THE FEED

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

HEADER:

a) Code	'WE'
b) Length	0x0B
c) Sequence Number	XXXX

ASCII DATA:

Not associated with any ASCII data.

TRAILER:
Checksum is not computed.

5. CONTACT INFO

Name	Email	Contact Number
DOTEX Business	**dotex@nse.co.in**	91-22-26598385
Technical Support	**infofeed_support@nse.co.in**.	-

6. CHEKSUM

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);
}
```

7. EXAMPLE: FUNCTION FOR DECOMPRESSION

```
lzo_decomp (char cInputBuf [], unsigned int ipLength, char cOutputBuf [],
unsigned *opLength, unsigned short * lzo_errorcode)
{
        int r;
        Char mess [50];
        r = lzo1z_decompress ((unsigned char *) cInputBuf,
ipLength,
                                (unsigned char *) cOutputBuf,
opLength,
                                 NULL);

        If ( r == LZO_E_OK)
        {
         Print (mess," Decompressed %lu bytes back into %lu bytes\n",
                                    (long) ipLength, (long) *opLength);

         Return true;
        }

        Else
        {
        OutputDebug ("Internal error - decompression failed");
        Return false;
        }
}
```

8. 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 traded 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 the data after transmission. DotEx 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 the security depending on the issuer like Central government (GS), state government (SG), public sector unit, institutions, banks, corporate, 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 repurchase 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 a particular security.

TRADE HIGH PRICE

Highest price at which trade taken place in a security.

TRADE LOW PRICE

Lowest price at which trade taken place in a 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 face value of all the securities that have been traded.

WEIGHTED YIELD

Weighted yield indicated for the traded security is calculated on the basis of 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 in day at 03:00 PM and 05:45 PM on normal trading days. This packet is sent to communicate market 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

Heart Beat is a packet that is sent at regular intervals through out 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
Suspended - S