

TECHNICAL DOCUMENT

5 MINUTES SNAPSHOT DATA

CURRENCY DERIVATIVES MARKET

01 DEC 2008



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

COPYRIGHT NOTICE

All rights reserved. No part of this document may be reproduced or transmitted in any form and by any means without the prior permission of DotEx International Ltd.

INDEX

CHAPTER	CONTENTS	PAGE No.
1.	INTRODUCTION	4
2.	CONNECTION DETAILS	4
3.	DATA DETAILS	5
	3.1 MARKET INFORMATION	5
	3.1.1 MARKET FILES	5
	3.1.2 OPEN INTEREST FILES	5
	3.2 SECURITY INFORMATION	5
	3.3 BHAVCOPY INFORMATION	5
4.	DATA STRUCTURE DETAILS	6
	4.1. MARKET INFORMATION	6
	4.2. OPEN INTEREST INFORMATION	6
	4.3. SECURITY INFORMATION	7
	4.4. BHAVCOPY INFORMATION	8
5.	CONTACT INFO	9
6.	NOTE	10
7.	DATE AND TIME CONVERSION	11

CURRENCY DERIVATIVES MARKET 5 MINUTES SNAPSHOT DATA

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 packet format, where as the delayed data and End of day data is available in the form of files.

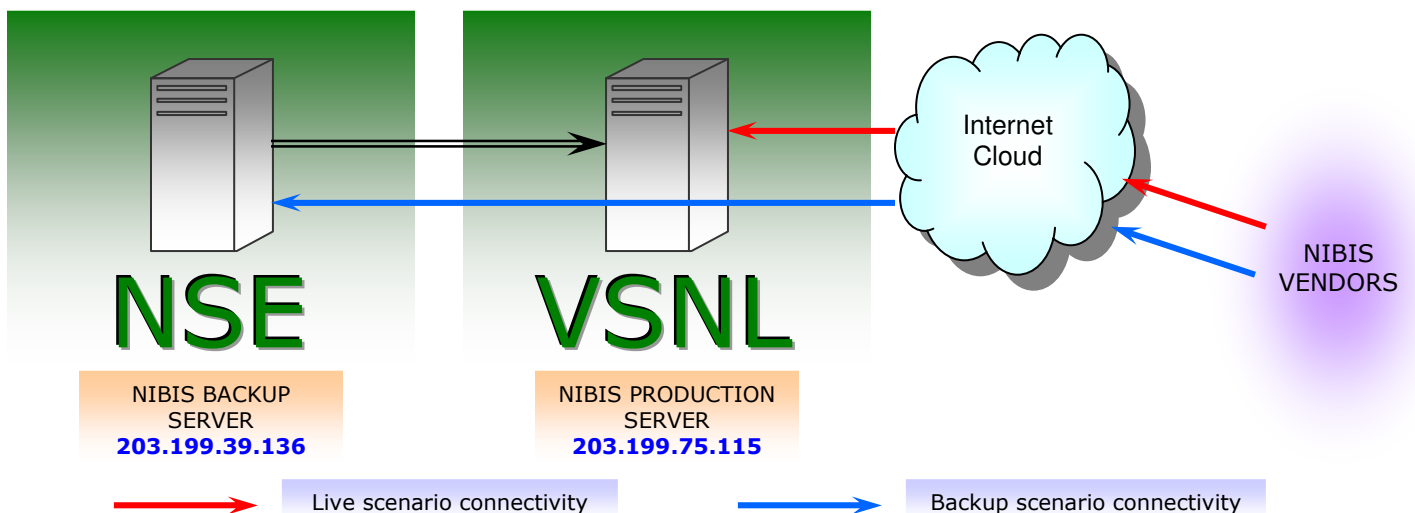
The Nibis server that caters the NIBIS vendors is available through internet. All Nibis vendors connect the server through internet and use FTP protocol to download the files. The files on this server are generated on regular 5 minutes interval basis. The vendors are provided with a User ID and password that is enabled for the agreement period.

2. CONNECTION DETAILS

The vendors connect the NIBIS server through Internet using FTP protocol. The production server IP address is 203.199.75.115 while the backup server IP address id 203.199.39.136.

DotEx also provides a backup setup of delayed data service. The backup server is always in disabled mode and it is enabled only in case of any hardware / other problem in production server. The backup server is located in NSE premises at BKC while Production server is located at VSNL, Prabhadevi. The files are initially generated on backup server and then instantaneously replicated on Nibis Production server.

Structural Diagram



3. DATA DETAILS

The Currency Derivatives Market (CDM) data files are generated in binary format at a regular interval of 5 minutes. These are *.mkt, and *.oi. The other data files generate on server are *.dat and *.txt. These entire files have categorised into different categories according to their data contents i.e. Market Information Files, Security Information Files, and Bhavcopy Information Files.

3.1 MARKET INFORMATION

The Market information data files are generated in /CDM05/DATA/"<Month>DDYYYY" folder on the server. i.e. *.mkt and *.oi.

3.1.1 MARKET FILES

The *.mkt (Where "*" stands for a numeric number) files contains market statistics and order information of the CDM contracts that are being traded in last 5 minutes along with 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 i.e. 09:00 hrs. To 17:30 hrs. These files are generated in incremental count number on a trading day starting from 1.mkt.

3.1.2 OPEN INTEREST FILES

The *.oi (Where "*" stands for a numeric number) files contains the details of open interest information of contracts that are traded during 5 minutes. These files are generated at regular interval of 5 minutes. These files are generated during normal trading period i.e. 09:00 hrs to 17:30 hrs.

3.2 CONTRACT INFORMATION

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

3.3 BHAVCOPY INFORMATION

The bhavcopy information file is generated at around 18:00 hrs in /CDM05/bhavcopy folder on each trading day. The file name is **CDMMKTSTATSYYYMMDD.TXT**. This file contains the End of the Day values of the contracts that are traded on that trading day.

4. DATA STRUCTURE DETAILS**4.1 MARKET INFORMATION**

FILE PATH - /CDM05/DATA/"<MONTH>DDYYYY"
 FILE NAME - *.mkt

FIELD	DATA TYPE / LENGTH	DESCRIPTION
HEADER		
Timestamp	LONG (4 Bytes)	Time when the record is updated
Message Length	Short Integer (2 Bytes)	Size of DATA packet
DATA		
Token Number	LONG (4 Bytes)	Unique identifier for contract
Market Type	CHAR (1 Bytes)	N-Normal Market
Best Buy Price	CHAR (17 Bytes)	Precision up to 7 decimal places
Best Buy Quantity	CHAR (12 Bytes))	In no of contracts
Best Sell Price	CHAR (17 Bytes)	Precision up to 7 decimal places
Best Sell Quantity	CHAR (12 Bytes)	In no of contracts
Last Traded Price	CHAR (17 Bytes)	Precision up to 7 decimal places
Total Traded Quantity	CHAR (12 Bytes)	In no of contracts
Average Traded Price	CHAR (17 Bytes)	Precision up to 7 decimal places
Security Status	CHAR (1 Bytes)	Blank for active and 'S' for suspended
Open Price	CHAR (17 Bytes)	Precision up to 7 decimal places
High Price	CHAR (17 Bytes)	Precision up to 7 decimal places
Low Price	CHAR (17 Bytes)	Precision up to 7 decimal places
Close Price	CHAR (17 Bytes)	Precision up to 7 decimal places

4.2 OPEN INTEREST INFORMATION

FILE PATH - /CDM05/DATA/"<MONTH>DDYYYY"
 FILE NAME - *.oi

FIELD	DATA TYPE / LENGTH	DESCRIPTION
HEADER		
Timestamp	LONG (4 Bytes)	Time when the record is

		updated
Message Length	Short Integer (2 Bytes)	Size of DATA packet
DATA		
Token number	LONG (4 Bytes)	Unique identifier for contract
Market Type	CHAR (1 Bytes)	Market Type
Open Interest	CHAR (12 Bytes)	In no of contracts

4.3 CONTRACT INFORMATION

FILE PATH - /CDM05/SECURITY/"<MONTH>DDYYYY"
FILE NAME - CONTRACT.DAT

FIELD	DATA TYPE / LENGTH	DESCRIPTION
HEADER		
No data in header		
DATA		
Token number	LONG (4 Bytes)	Unique identifier for contract
Instrument Name	CHAR (7 Bytes)	Instrument Name
Symbol	CHAR (11 Bytes)	Symbol
Series	CHAR (3 Bytes)	Series
Expiry Date	LONG (4 Bytes)	Expiry Date
Strike Price	LONG (4 Bytes)	Strike Price
Option Type	CHAR (3 Bytes)	Option Type
Issue Start Date	LONG (4 Bytes)	Issue Start Date
Issue Maturity Date	LONG (4 Bytes)	Issue Maturity Date
Board Lot Quantity	LONG (4 Bytes)	Board Lot Quantity
Tick Size	LONG (4 Bytes)	Tick Size
Security Name	CHAR (26 Bytes)	Security Name
Record Date	LONG (4 Bytes)	Record Date
Ex Date	LONG (4 Bytes)	Ex Date
No Delivery Start Date	LONG (4 Bytes)	No Delivery Start Date
No Delivery End Date	LONG (4 Bytes)	No Delivery End Date
Book Closure Start Date	LONG (4 Bytes)	Book Closure Start Date
Book Closure End Date	LONG (4 Bytes)	Book Closure End Date
Remarks	CHAR (26 Bytes)	Remarks

4.4 BHAVCOPY INFORMATION

FILE PATH - /CDM05/BHAVCOPY/"<MONTH>DDYYYY"
 FILE NAME - CDMMKTSTATYYYYMMDD.TXT

FIELD	DATA TYPE / LENGTH	DESCRIPTION
HEADER		
No data in header		
DATA		
Instrument Name	CHAR (6 Bytes)	Instrument Name
Symbol	CHAR (10 Bytes)	Symbol
Expiry Date	CHAR (11 Bytes)	Expiry Date (DD-MM-YYYY)
Strike Price	CHAR (10 Bytes)	Precision up to 2 decimal places
Option Type	CHAR (2 Bytes)	Option Type (FF)
Market Type	CHAR (1 Bytes)	N - Normal Market
Open Price	CHAR (17 Bytes)	Precision up to 7 decimal places
High Price	CHAR (17 Bytes)	Precision up to 7 decimal places
Low Price	CHAR (17 Bytes)	Precision up to 7 decimal places
Close Price	CHAR (17 Bytes)	Precision up to 7 decimal places
Last Traded Price	CHAR (17 Bytes)	Precision up to 7 decimal places
Previous Close Price	CHAR (17 Bytes)	Precision up to 7 decimal places
Settlement Price	CHAR (17 Bytes)	Precision up to 7 decimal places
Total Traded Quantity	CHAR (12 Bytes)	Total Traded Quantity
Total Traded Value	CHAR (25 Bytes)	Precision up to 2 decimal places
Current Open Interest	CHAR (10 Bytes)	In no of contracts
Change In Open Interest	CHAR (10 Bytes)	In no of contracts

5. CONTACT INFO

Following are the contact details for business assistance:

Name	Email Address	Contact Numbers
Mr. Ved Malla	vmalla@nse.co.in	91-22-26598385
Ms. Prachee Chavan	pracheec@nse.co.in	91-22-26598385
Mr. Pankaj Agarwal	pagarwal@nse.co.in	91-22-26598385

You can also email us on **dotex@nse.co.in**

For technical assistance email us on **infofeed_support@nse.co.in**.

6. NOTE

- The token number (Symbol, Instrument name, Expiry date & Strike price and Option type) uniquely identify each contract. The vendor shall be provided with a binary file (**CONTRACT.DAT**) giving the combinations of all contract valid on the Exchange. Any further additions, modifications and deletions shall be sent to the vendors as part of the feed.
- All contract added to the Exchange's database will be updated in the file **CONTRACT.DAT**. Vendors are requested to please update their databases as per details in these files.
- EOD market statistics data file is a text file generated with **CDMMKTSTATSYYYYMMDD.txt** naming convention that contains the end of day data that is available at 18:00 Hrs. on each trading day.
- Prices field will have the precision up to 7th decimal places.

7. DATE AND TIME CONVERSION

- 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 interger. 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 : INoInput - 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;
}

```

- Sample program for convert long time in to HH:MM:SS format

```

/*****

```

Routine Name : TimeConv

Synopsis : This routine is responsible for processing a time (LONG) which constitutes a six digit interger. The integer is then converted into the following:
 (takes into account decade 70)
 a string - 99:99:99
 (defined in crack.h)

If the incoming number is zero it will return blank time

Parameter descriptions : INoInput - Incoming number
 cTimeStr - Time output in string format

Return value : NONE

```

*****/

```

```

VOID TimeConv(LONG INoInput,CHAR* cTimeStr)

```

```

{
    struct tm * pTimeStruct ;

```

```
        CString          szTime;

if ( INoInput == 0 )
{
    strcpy(cTimeStr, "");
}
else
{
    pTimeStruct = localtime ( ( time_t * ) &INoInput ) ;

    if ( pTimeStruct->tm_isdst == 1 )
        --pTimeStruct->tm_hour ;

    if (pTimeStruct->tm_hour == -1 )
    {
        pTimeStruct->tm_hour = 23; // make sure that from midnight
                                   returns to 11pm
    }

    sprintf ( cTimeStr ,
              "%02d:%02d:%02d" ,
              pTimeStruct->tm_hour ,
              pTimeStruct->tm_min ,
              pTimeStruct->tm_sec
            ) ;
}
}
```