**Algo Dashboard Phase 1**

**Note:-**

1. All tables are created in database *GRG\_DB.*
2. Project is stored at /else/psgalgo/AlgoDashboard/ in server 10.250.22.33
3. All the input config files should be in folder ./input/Phase1/
4. For loading raw data directly from csv , it should be named RawData.csv and kept in input/Phase1/
5. All the input bhav copies arer taken from /else/home/oaa/Bhavcopy/
6. All the input algo terminal list CSVs should be in folder input/Phase1/Algofiles/
7. Inside the config file, names should be complete names(including extensions)
8. All output csvs will be located at output/Phase1/{date}/

Praccont.java :-

See previous documentation for implementation details.

Basically we create a server vs lob mapping and a list of algo terminals .

Prac.java :-

See previous documentation for implementation details.

Here we create a table Consolidate\_vol\_data conataining orders,id, timestamp,terminal,price,quantity,etc,. corresponding to trade on specific date (given as input) from platform *Omnesis*.

PreRequirements.java :-

In this module we combine data from Consolidate\_vol\_data and generic platform-SBU data from different database *GENERIC\_OTA* ‘s table Global.TradeDetailsMainTable to form table RawData.

Important thing to note is SBU names from *Omnesis* and its exchange-segment names are by default capitalized in final RawData and specific format is required to be given as input in new generic data exchange-segment.

Exchanges can be NSE,BSE etc,.

Segments are CASH,CDS,FO etc,.

Instrument are FUT,OPT ,blank etc,.

Exchange-segments are in format ExchangeName\_SegmentName Eg. NSE\_FO.

Following is the list of standard SBUs in Omnesis.

1. COMMODITIE
2. FIC
3. IE
4. MARKETS
5. FIM
6. TREASURYM

Blank sbu names ar given a default value ABC

NOTE:-As long as above format in Omnesis doesn’t change, code will produce accurate results.

All currency denomination in table created is in INR.

Also the terminal-sbu mapping from table RMS\_9\_DLS\_TERMINAL\_LIMITS from database *RMS\_VALIDATION*  is used for mapping Omnesis data to RawData.

Functionality to differentiate Algo terminals from NonAlgo ones is done in this module. For this user has to create a config\_Algo.txt file for providing methods to extract algo terminals. This process can be done in three ways

**Syntax of config\_Algo.txt :-**

**TEXT sbuName Keyword** // Keyword is the text to search for in all terminals of given sbu .

**PLATFORM sbuName platformName** // to specify that all terminals that fall in this sbu and platform as algo ones.

**FILE filename.csv** // to select all terminals from given file and mark them as algo ones

Note:- config selection is only for Omnesis platform, in generic table there is a column that differentiates algo and nonalgo terminals

Requirements.java :-

In this module we create two tables from use of RawData namely :- PlatformSpecific and ExchInsSpecific .

PlatformSpecific contains total and algo turnover for specific {platform, sbu, exchange-segment}

ExchInsSpecific contains total and algo turnover for specific {instrument,sbu, exchange-segment}

Turnover denomination is in crores .

Table.java :-

In this module we create the final output tables in database *GRG\_DB* .

All the final tables use data only from PlatformSpecific and ExchInsSpecific .

dbo.SBUAlgoVol  Table **1A**

dbo.PlatformAlgoVol Table **1B**

dbo.SBUAlgoMarket  Table **2A**

dbo.PlatformAlgoMarket Table **2B**

dbo.SBUAlgoTotal Table **3A**

dbo.PlatformAlgoTotal Table **3B**

dbo.EdelweissExchAlgoVol Table **4**

dbo.ExchSegAlgoTotal Table **5**

dbo.ExchInsAlgoTotal Table **6-7**

dbo.Exchange\_vs\_EWExchAlgo Table **8**

dbo.SBUWise\_Exchange\_vs\_EWExchAlgo Table **9**

Table numbers are corresponding to the project statement.

Read the comments in code. There I have explained all the hash maps and table fields and the formulae if any to calculate the percentage.

Also in this module we read config\_Bhav.txt to extract information from bhavcopies.

**Syntax of config\_Bhav.txt :-**

**ExchangeSegmentName bhavcopyCSVName dateFormat Multiplefactor SeriesColumnNumber TurnOverColumnNumber**

//ExchangeSegment name should be corresponding to format mentioned in prerequirements.java doc above.

//bhavcopyname should not contain .csv extention and a “|” is to inserted where date is to appear. Eg. fo|bhav

//date formats are fixed.

1.ddMMMyyyy 2. ddmmyy 3. ddMmmyyyy

//Multiplier factor denotes the denomination of turnover column

Eg. 100000 if calculated in lakhs

//Series columnNumber is specifically for NSE\_CASH to select only “EQ” series turnover

In its implementation I have deliberately inserted blank rows in tables for easy readability.

Output.java :-

This class is basically for printing all sql tables in database GRG\_DB created in above files in their respective csv files.

In this printing of tables null fields are replaced by blanks

LoadRawData.java :-

This class is for uploading RawData directly from csv file to sql table. Without actually going through process of running Prac.java , Praccount.java and PreRequirements.java .

Format of given RawData.csv columns:-

Platform, SBU, ExchangeSegment, Instrument, TransactionType, StrikePrice, FilledQuantity, Price, isAlgo, TurnOver

Execution:-

* + Connect to server 10.250.22.33 .
  + Change Directory to /else/psgalgo/AlgoDashboard/
  + Run the command “sh phase1start1.sh” for taking data from databases.
    - * + OR
  + Run the command “sh phase1start2.sh” for taking data from RawData.csv