Factotum RMS

Version 1.0.3

Release Date: 26.10.2021

Risk Management System (RMS)

1. Introduction

Risk Management System (RMS) is a real time surveillance engine that carries a list of predefined checks to minimize the losses at client level and system level. Checks are *defined values* based on the calculated risk of a position or strategy. Breach of a *defined value* will pause a position and make the order void. The order engine and client terminal (Fractal) are connected through the RMS. The RMS is a server software application that processes the received order through the client interface, and eventually, the validated order is sent to the order engine. RMS checks the correctness of client positions and synchronizes them with that of the order engine. RMS stores all the information in a database which is used for validating client positions and orders. RMS works on the principle of TCP (Transmission Control Protocol) based client-server methodology. Only the RMS administrator has the authorization to connect to the RMS. RMS also records the history of the orders and executed trades by all individual client terminals.

Each order is validated as per the following mentioned checks before passing it to the trading system. The definition of each of the checks has been juxtaposed in right.

- **Max Positions:** Maximum number of rows that can be applied by the user.
- **Max Qty Per Order:** Maximum quantity of trades allowed in each order per user.
- **Max Lots per Position:** Maximum number of lots allowed per position for each user.
- **Max Lots per Order:**Maximum number of lots allowed for a single order per position for each user.
- **Max Value Per Order:** Maximum value permissible for user in single order.
- **Max Loss Per Position:** Maximum loss permissible for user per position.
- **Max Loss Per Day:** Maximum loss permissible for user per day.
- **Net Option Value:** Maximum net option value permissible per user.
- **Net Future Value:** Maximum net future value permissible per user.
- **Net Value:** Maximum net value permissible per user.
- **Max Position Buy Limit:** Maximum number of open position buy quantity allowed per person.
- **Max Position Sell Limit:** Maximum number of open position sell quantity allowed per person.
- **Max Cumm Value:** Maximum cumulative value permissible per user.
- **Index Level Change:** Maximum index change (in percentage) allowed per user. Positions are paused if the index limit is breached.
- **Freeze Quantity check:** Maximum order quantity can not be more than freeze quantity (This check is inbuilt in the RMS).
- **MWPL check:** Maximum positions can’t be more than MWPL limits. (This check is inbuilt in the RMS).
- **Security value limit:** Maximum security value can’t be more than the defined limits. These limits are defined in a separate file.

1.1. Functionalities of RMS:

The RMS has three major functionalities.

1. User level administration.
2. Setting limits/exposures and versatile checks.
3. Live risk monitoring of all trade positions.

1.1.1. User level administration

At user level administration, RMS defines the login parameters of each client. RMS also controls the trade position of each client in regard to risk management.

1.1.2. Setting limits/exposures & deploys checks

Defining a trader's limits and exposures in his/her respective trade positions is also done through the RMS interface. Some of the important checks/limits that can be deployed by using the RMS are:

- *Maximum loss per position* and *maximum loss per day* can be managed from the console of RMS.
- *Turnover limit* and *maximum lots per order* can also be defined.
- The RMS displays consolidated view of user positions and the associated marked to market (M2M) values.
- Administrator can increase or decrease position/trade limits of the user as per his/her risk calculations.
- RMS also gives the administrator the right to view order books, trade books, and activity logs of individual clients.

1.2. Live risk monitoring of trade positions

RMS can monitor *marked to market* (M2M) as well as the *realized loss or profit* (RLP) of every user. The monitoring can be done feasibly using the consolidated view window of Factotum RMS module. The RMS can also take appropriate measures whenever the set values are breached by any user terminal. Some of the risk measures are being discussed in the following section:

1. **Risk management for order quantity:** RMS is configured in such a way that it will not fire any order exceeding the limits mentioned in its checks.

2. **Price should be in the range:** System will not fire any order at a price greater than DPR (day price range) value.

3. **DPR:** *Day Price Range (DPR)* defines the upper and lower limit of the price of any stock or instrument on an intraday basis and the DPR is defined by the respective exchanges (i.e. NSE, BSE, MCX etc.).

4. **Risk management for order value:** The RMS will prevent any order from being executed, whenever the order value (quantity multiplied by price) is greater than the maximum value defined by RMS.

5. **M2M (marked to market) position; Net position profit or loss (P/L) should not cross the available margin as per RMS rules:** RMS continuously scrutinizes the positions of each client. Occurrence of instances when the M2M profit or loss exceeds the defined limits, enable the RMS to pause the user and notifies the administrator.

6. **Margin limit:** RMS limits the client to initiate position as per his/her margin. RMS validates each order with his/her earlier built-up position. If current order value and the earlier position values exceeds the '*Max. Position Value'* defined in RMS, then orders placed will get rejected, i.e. will not be executed. Any measure or action taken as such is always notified to the respective RMS administrator.

7. **Row limit:** RMS sets the maximum number of rows allowed per client**.** System will not allow more rows than the defined limits in the RMS.

8. **Cumulative open order limit check:** The cumulative value of all unexecuted orders placed from the algo-trading terminals is limited to the threshold set by the member/administrator at individual client level. *Cumulative open order value* for a client is the total value of its unexecuted orders released from the trading member's system. The total value of unexecuted orders is checked by the RMS.

9. **Freeze Quantity check:** *Freeze Quantity* for a contract is defined in the contract file and orders are not released whenever the value defined for freeze quantity is exceeded.

10. **MWPL check:** Market Wide Protection Limit (MWPL) limits are defined in ***oi_cli_limit.csv*** file which we download from the exchange (NSE) website/ftp server. We update the oi_cli_limit.csv file from exchange in every 30 mins.

11. **Security value limit:** Orders are not released in breach of defined security value limit. To define limits, ***security_value_limit.csv*** file is placed in NebulaFNORMS folder and it can be updated anytime even in the live market.

2. RMS Setup and Login

Factotum RMS and clients are platform independent and can be installed easily.

2.1 Prerequisites

1. Install postgres 9.3 version
2. Pgadmin3
3. Java 1.8 (32 bit)
4. *Putty (needed if setup is on windows)
5. *winscp (needed if setup is on windows)
6. Contract files (FnO or Currency as needed)

FnO contract files:	*Currency contract files:*
fo_contract_stream_info.txt contract.txt fo_spd_contract_stream_info.txt spd_contract.txt fo_secban.txt cm_contract_stream_info.txt security.txt Bhavcopy file (eg. fo22OCT2019bhav.txt)	cd_contract_stream_info.txt cd_contract.txt cd_spd_contract_stream_info.txt cd_spd_contract.txt cd_participant.txt fo_secban.txt cm_contract_stream_info.txt security.txt Bhavcopy file (eg. fo22OCT2019bhav.txt)

2.2 Setup FactotumRMS

For Windows

We need to place all the contract files in the folder : C:\NebulaFNORMS.

For Linux

We need to place all the contract files in the folder : /root/NebulaFNORMS.

** Apart from the mentioned folders we need to create 2 more folders:

1. 1st in NebulaFNORMS folder containing folder name same as RMS server port.
   Eg: C:\NebulaFNORMS\8710

2. 2nd in the newly created folder for report generation.
   Eg: C:\NebulaFNORMS\8710\Reports

2.3 Start FactotumRMS:

To run the RMS, the steps required to be followed by the RMS admin are to be followed sequentially as per below given instructions:

1. Install 32 bit java and Pgadmin in the system.
2. Run the RMS jar file.
3. Login with the default credentials:
4. Username: *user0*
5. Password: 12345678

Connect RMS with the Core (if the Binary is running on the Linux Machine).
Note: RMS is compatible to run both in a Windows or Linux based OS.

Path of the RMS for Windows: *file:C:\NebulaFNORMS\8702*
Path of the RMS for Linux: */root/NebulaFNORMS/8702*

Contract files are needed to be placed in NebulaFNORMS directory.
Reports directory should be placed in 8702 directory where contract-wise trade logs are generated.

8702 is the directory in which our RMS jar file and report directory is located. We can run the RMS with simple java (32 bit) installed in the system.

2. Trading System:

Trading System comprises the following three parts:

- **Client:** It is also called Trading Terminal and is used by traders for trading purposes.
- **RMS:** It is used by the administrator for risk management & trade monitoring purposes.
- **Core:** It is the order engine that fires trades from the NSE.

3. Description of different parts of the RMS:

RMS consists of the following parts which has been described below:

3.1 Dashboard
3.2 Settings
3.3 Users Management
3.4 Algo Management
3.5 System View
3.6 System Checks
3.7 User Stats
3.8 RMS Reports
3.9 RMS Stats

3.1. Dashboard :

From the *Dashboard* area, we can check whether our RMS is running smoothly or not by looking over its system statistics.

Login Steps from Dashboard:

- **Archiving data**
- **Recovering data**

3.1.1 Archiving data

The data-archiving should be done once at the start of the day to delete all trades of the previous day and start the day with new trade data.

Steps to archive data are given as follows:

Archive button -> Login Server -> Refresh system state

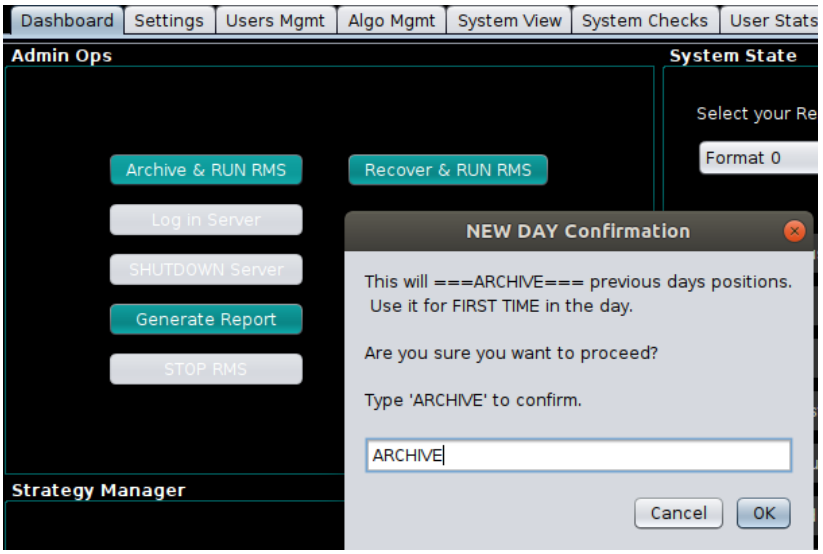


Fig. Archiving trade

Note: Archiving must be done once in a day, because once the data has been archived, then it will not be available for the recovery.

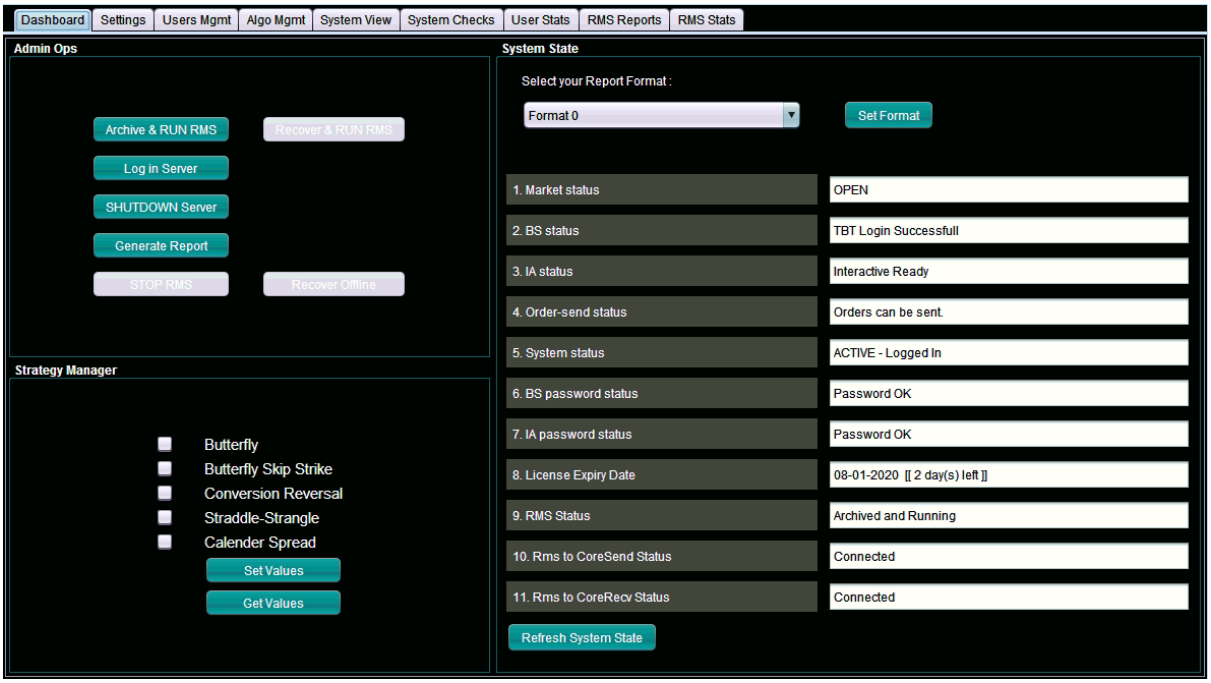


Fig.1: User Interface of the *Dashboard* panel

3.1.2. Recovering data

In order to recover data, the following steps are needed to be followed by the user.

Click on *Recover* -> Login to the *Server* -> Refresh system state

The *recovering data* functionality is used to recover the saved positions of the clients in case the system crashes or stops by any reason.

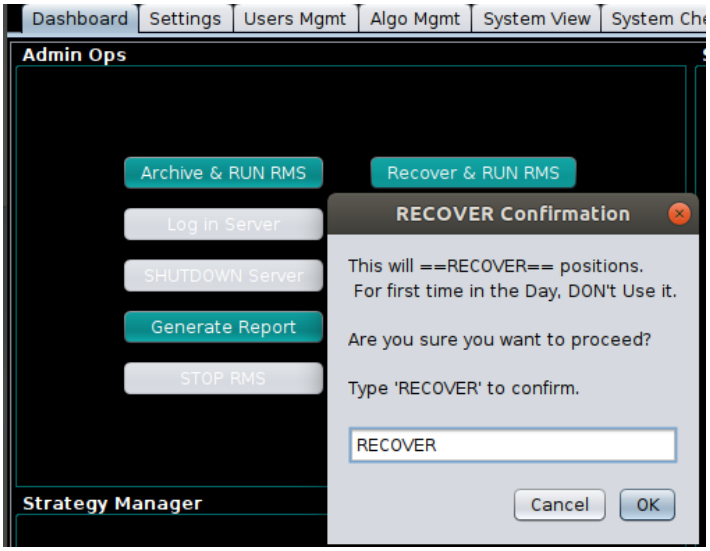


Fig.1.a Recovering trade

3.1.3 System stats:

The following table illustrates the different *system states* of the RMS.

System state	Status displayed
Market Status	: Open
BS Status	: Tick By Tick (TBT) Socket Created
IA Status	: Interactive Socket Created
Order-sent Status	: Orders can be sent
System Status	: Active Logged in
BS password status	: Password OK
License Expiry Date	: [[168 days left]]
RMS Status	: Archived and Running
Rms to CoreSend Status	: Connected
Rms to CoreRecv Status	: Connected

3.2. Settings

In the setting section, the stream and port IPs can defined and should resemble the below figures:

Stream #1 IP:: 239.70.70.41 & Stream #1 port:: 17741
Stream #2 IP:: 239.70.70.42 & Stream #2 port:: 17742

Stream interface, i.e. the linux machine interface IP to be filled with the following given details:

Interactive IP ADDRESS:: 172.19.13.85 & Interactive Port :: 10825
Recovery IP:: recovery ip of machine & Port :: 2687
IP address:: linux machine IP Port :: 8723(default)

Server password is needed to be set for starting the RMS. Steps to be followed to set the server password are:

Set server password -> IA -> Neat@FO1 (using default password) -> set new password -> Done

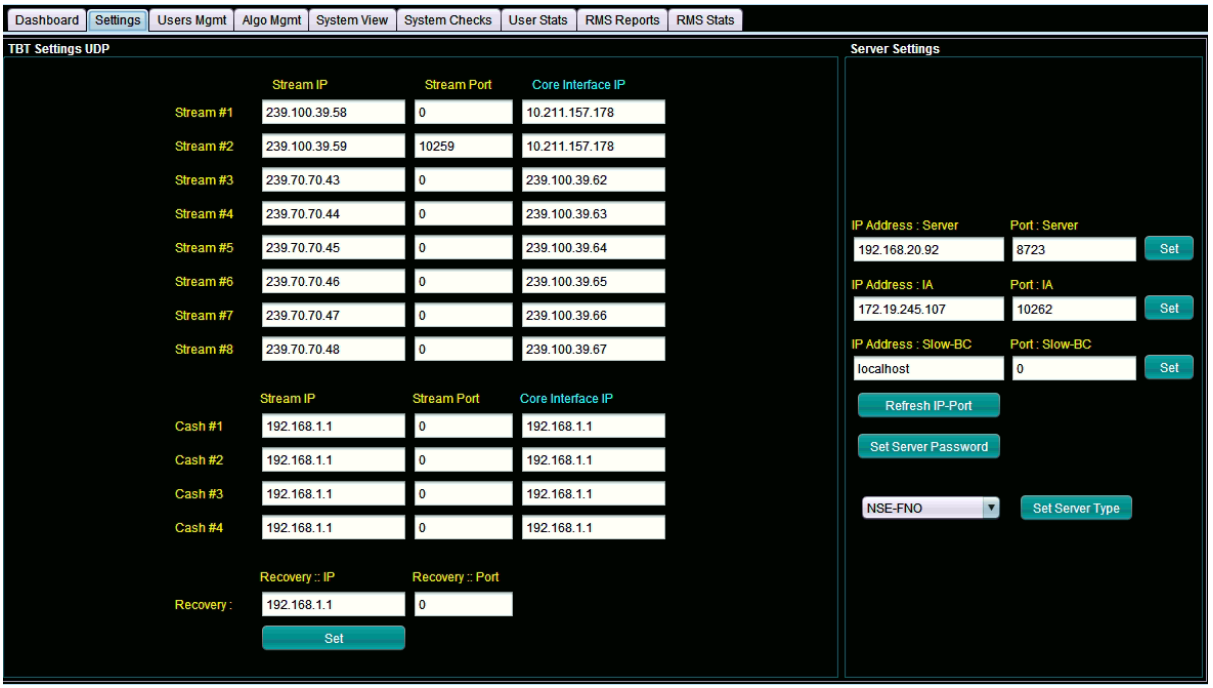


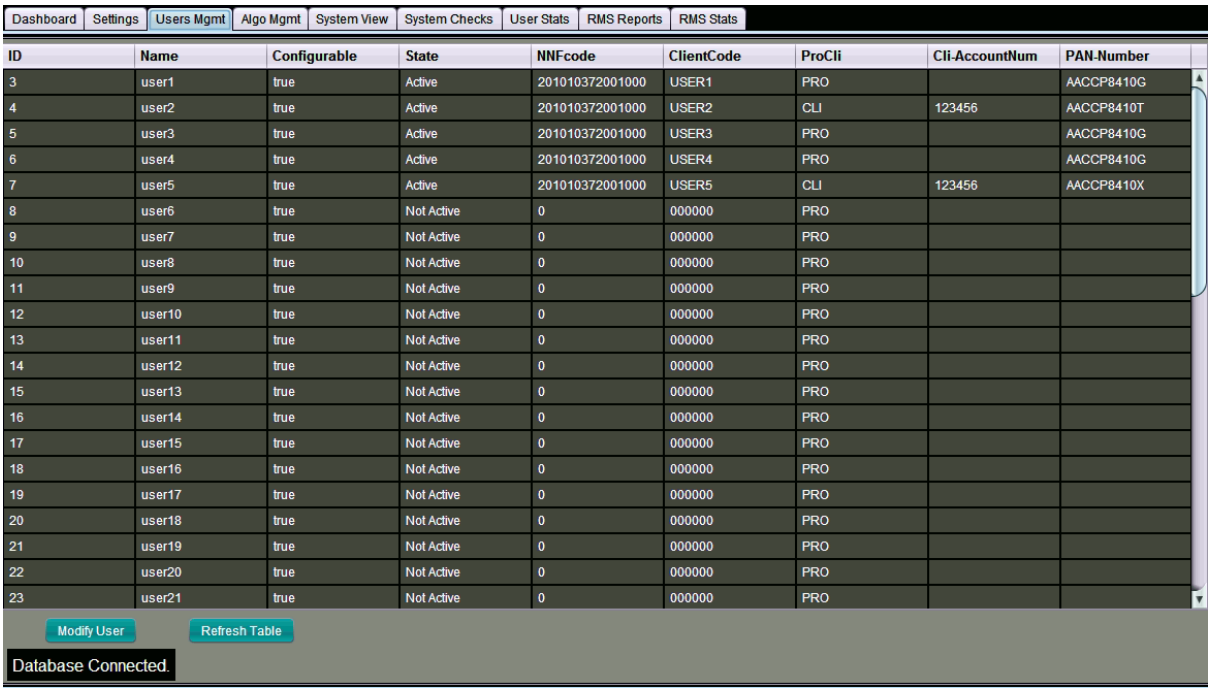
Fig.2. User interface of the *Settings* panel

3.3.User Management:

In the *User Management* window, the admin can fill-up all the details of users and modify it as active or inactive. If the user is trading, then the status can be defined as *Active*, and when the user is not-trading, then the status assigned to hi/her is *Inactive*.

In the user management, the admin should fill the following information of the users in below stated manner:

Name : User1
Configurable : True
State active : Active
NNF CODE : 201210001001000(15 Digits)
Client code : CTCL1
Pro cli : Pro or Cli
Cli account number : Required in case of client
Pan account number (REQUIRED) : It is mandatory



ID	Name	Configurable	State	NNFcode	ClientCode	ProCli	Cli-AccountNum	PAN-Number
3	user1	true	Active	201010372001000	USER1	PRO		AACCP8410G
4	user2	true	Active	201010372001000	USER2	CLI	123456	AACCP8410T
5	user3	true	Active	201010372001000	USER3	PRO		AACCP8410G
6	user4	true	Active	201010372001000	USER4	PRO		AACCP8410G
7	user5	true	Active	201010372001000	USER5	CLI	123456	AACCP8410X
8	user6	true	Not Active	0	000000	PRO		
9	user7	true	Not Active	0	000000	PRO		
10	user8	true	Not Active	0	000000	PRO		
11	user9	true	Not Active	0	000000	PRO		
12	user10	true	Not Active	0	000000	PRO		
13	user11	true	Not Active	0	000000	PRO		
14	user12	true	Not Active	0	000000	PRO		
15	user13	true	Not Active	0	000000	PRO		
16	user14	true	Not Active	0	000000	PRO		
17	user15	true	Not Active	0	000000	PRO		
18	user16	true	Not Active	0	000000	PRO		
19	user17	true	Not Active	0	000000	PRO		
20	user18	true	Not Active	0	000000	PRO		
21	user19	true	Not Active	0	000000	PRO		
22	user20	true	Not Active	0	000000	PRO		
23	user21	true	Not Active	0	000000	PRO		

Fig.3: User interface of the *Users Management* panel

3.4. Algo Management:

It is very critical and important to fill the Algo-management window fields correctly. Algo-ID and Algo-category are needed to be filled in the Algo management window. In case of a test machine we can fill-in any random number or ID in the *Algo management* window.

In case of wrong information input in any of the algo-management fields, a penalty might be levied by the respective exchange according to their respective trading rules and policies.

Name	Code	Algo-ID	Algo-Cat
IOC-2Leg	210	99999	2
IOC-3Leg	310	99999	2
IOC-4Leg	410	99999	2
IOC-2Plus1	501	99999	2
IOC-5Leg	555	99999	2
IOC-6Leg	666	99999	2
IOC-FFS	810	99999	2
IOC-SF	820	99999	2
Manual-1Leg	10101	99999	2
Spread-2Leg	10102	99999	2
1Leg-Momentum	10150	99999	2
IOC-3Leg	10151	0	0
BID-2Leg	10201	99999	2
BID-3Leg	10301	99999	2
BID-3Leg-CRSpecial	10333	99999	2
BID-4Leg	10401	99999	2
BID-3Leg-BoxSpecial	10444	99999	2
BID-5Leg	10501	99999	2
BID-6Leg	10601	99999	2
BID-FFS	10810	99999	2

Fig.4: User interface of the *Algo Management* panel

3.5. System View

This window tells the user about various counts related to the orders, trades or errors. Users can also follow this window to check if their system is working in the desired manner.

Field description are as follows:

Invitation Count	: [ ]
Current Order Number	: [ ]
Orders Sent To Exchange	: It tells how many orders have been sent to exchange
Orders Dropped from adapter	:In case the core generates more order-packets than the size of the system's message line, then the packets will be dropped by the engine in order to maintain the message-line functional
Orders confirmed	: No. of confirmed orders
Orders Cancelled	: No. of cancelled orders
Orders Traded	: No. of traded orders
Orders Errored	: No. of orders with error
TBT Messages Received from exchange	: No of tick-by-tick packets received
TBT Messages Accepted	: No. of positions added to the core
Number of Positions in Server	: Number of positions of trade
IA Sequence Number	: Heart beats of core machine

NOTE:

1. **Orders Cancelled** should be less than **Orders Confirmed**. Please kill the binary and RMS and reach out to the developer team.

2. Please keep an eye on the no. of **Orders Errored**. All rejected orders consume the message line and give up the opportunity of other good orders.

3. If **TBT Messages Accepted** is not increasing even after adding a new position then contract files in RMS must be checked.

4. On a successful BOD, the value of **TBT Messages Accepted** should be **greater than 10. In other cases please contact the developer team.**

3.6 System Checks

3.6.1. System level checks

Orders per second: No. of orders fired per second
Max Positions: Shows the total max positions.
Max Cumm Value: Maximum Cumulative value allowed. Its maximum value set can be 10 Crore.
Max net Value: Maximum Net Value Permissible to System.
Max option net value: Max option value for the System.
Max future net value: Max future value for the System.
Max Loss: Maximum loss Permissible for the system.

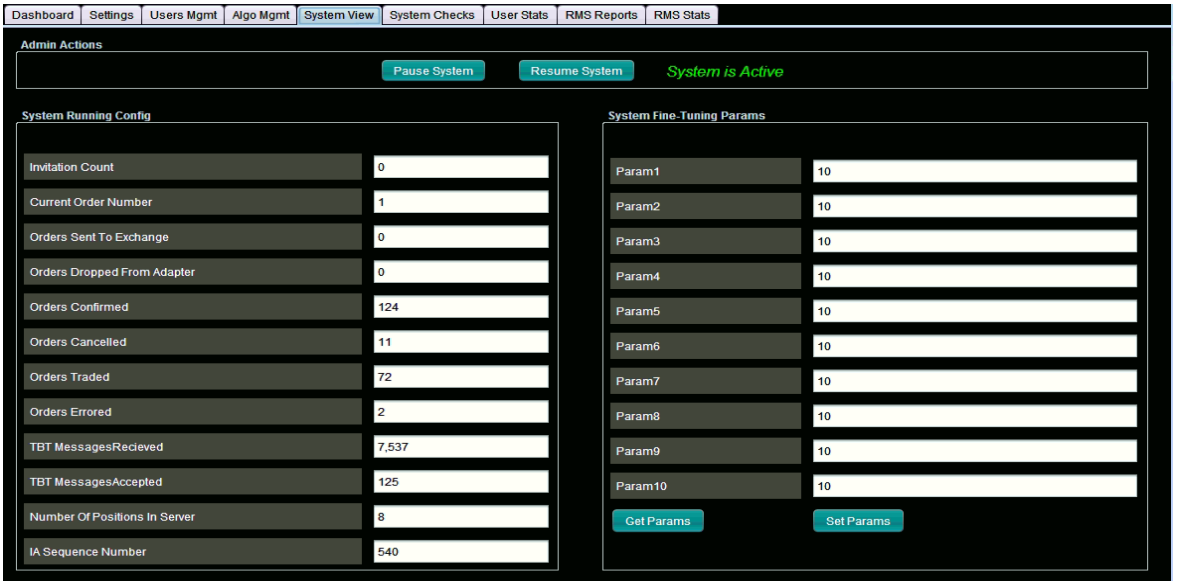


Fig.5: User interface of the *System View panel*

3.6.2. User level checks

Max Positions: Maximum number of rows that can be applied by a user.
Max Qty Per Order: Maximum Quantity allowed in each order by a user.
Max Lots Per Position: Maximum no. of lots allowed per position for a user.
Max Lots Per Order: Maximum number of lots allowed in a single order per position for a user.
Max Value Per Order: Maximum Value permissible for user in single order.
Max Loss Per Position: Maximum loss permissible for user per position.
Max Loss Per Day: Maximum loss Permissible for users in a day.
Net Option Value: Maximum Net Option value Permissible to a user.
Net Future Value: Maximum Net Future Value Permissible to a user.
Net Value: Maximum Net Value Permissible to a user.
Freeze Qty: Freeze qty for a contract is defined in the contract file.
MWPL check: These limits are defined in oi_cli_limit.csv which we can download from nse.
Security value limit: These limits are defined in security_value_limit.csv file i.e. placed in NebulaFNORMS directory.
Max Cumm Value: Maximum Cumulative value allowed. Its maximum value set can be 10 Crore.
Max Turnover Value: Maximum Turnover value allowed. Its maximum value allowed is (4 * Max Cumulative Value).
Max Margin: Maximum allowable limit to User
Index Level Change: Maximum index change (in percent) allowed for a user.
Positions are paused if the index set limit is crossed.
Max price Protection value (in percent): Maximum price protection value can be set for a user.Positions are paused if price protection value exceeds the set limit.

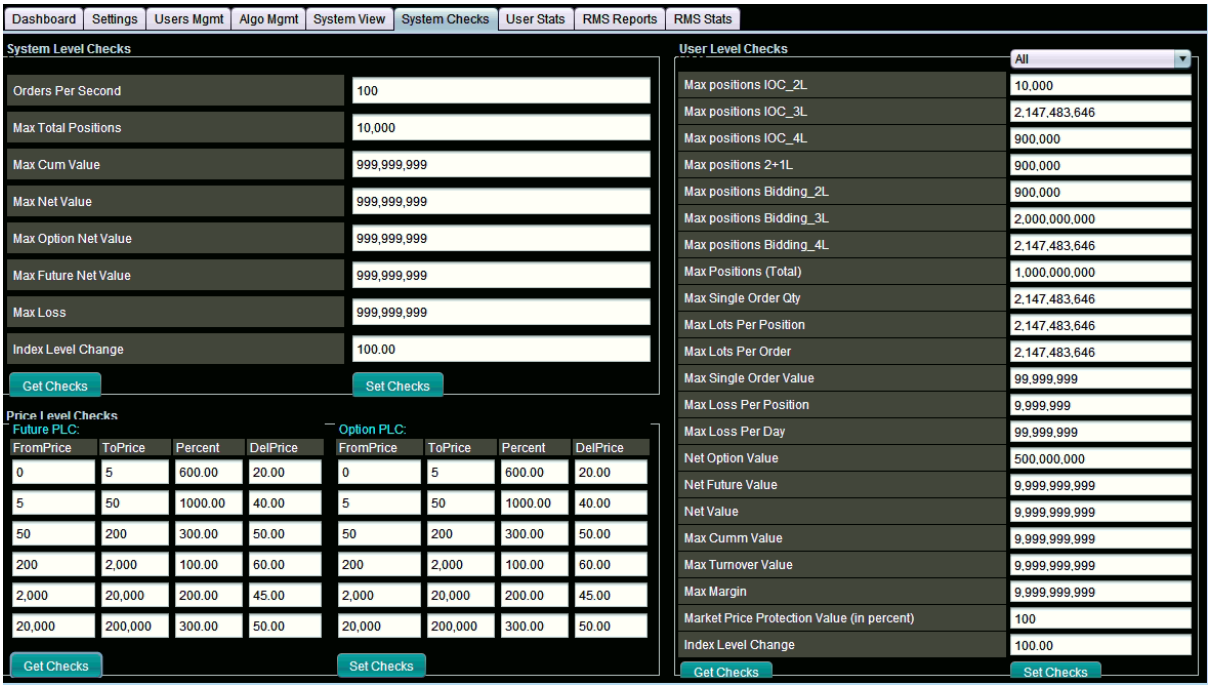


Fig.6: User interface of the *System Checks panel*

3.7 User Stats

In the *User Stats* window, we can see the users who are logged-in as well as their trading status.Also we can resume, pause, and reset the password of any user from this window.

Fig.7: User interface of the *User Stats window*

3.8 RMS Reports:

In this section we can generate the reports of all users or any specific user on the basis of the trade done by the trader or clients.

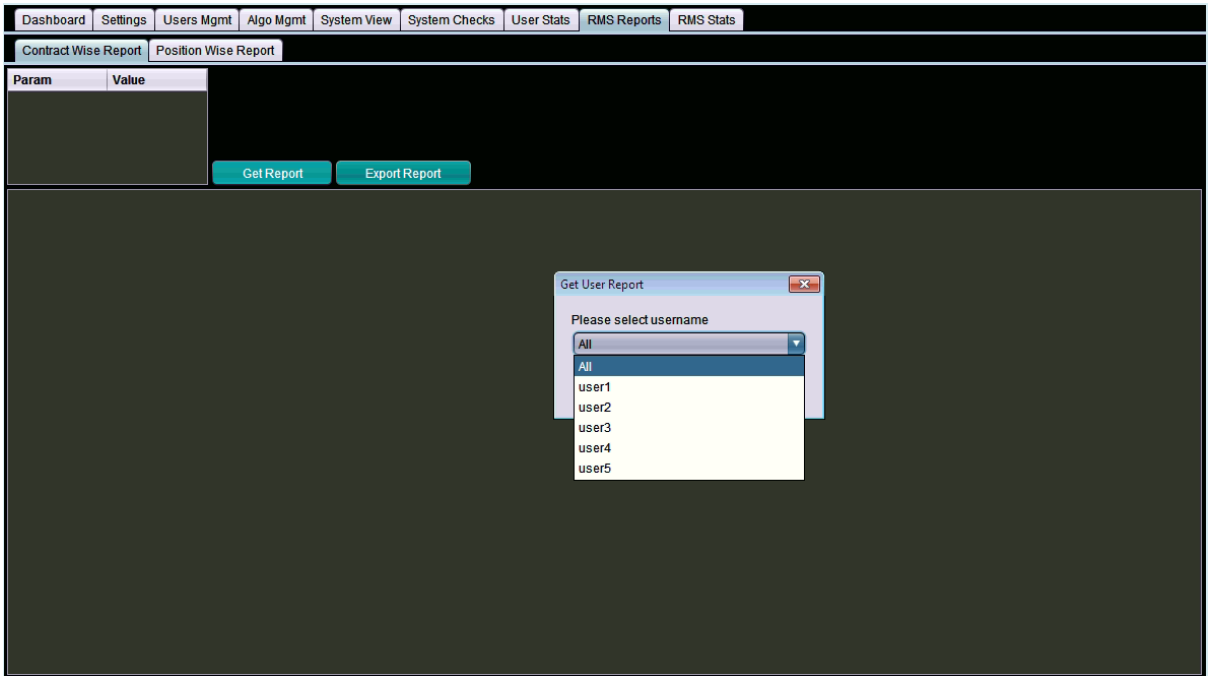


Fig. 8: User interface of the *RMS Report window*

To generate report *user-wise*, the following steps are needed to be followed:

Get report -> Select user -> user2 -> ok.

User report generation steps:

Get report -> Select user -> all -> ok.

3.9 RMS Stats:

Here we show *RMS Graphs*, *Position Stats*, and *Order Trade Stats*.
RMS graphs:Here we used to show Line Usage Vs Time (in second) for a particular machine
Order-Trade Stats: In this section we have shown the Pgid, Orders Count, Trades, Order/Trade Ratio, Reject Count, Order rate etc.

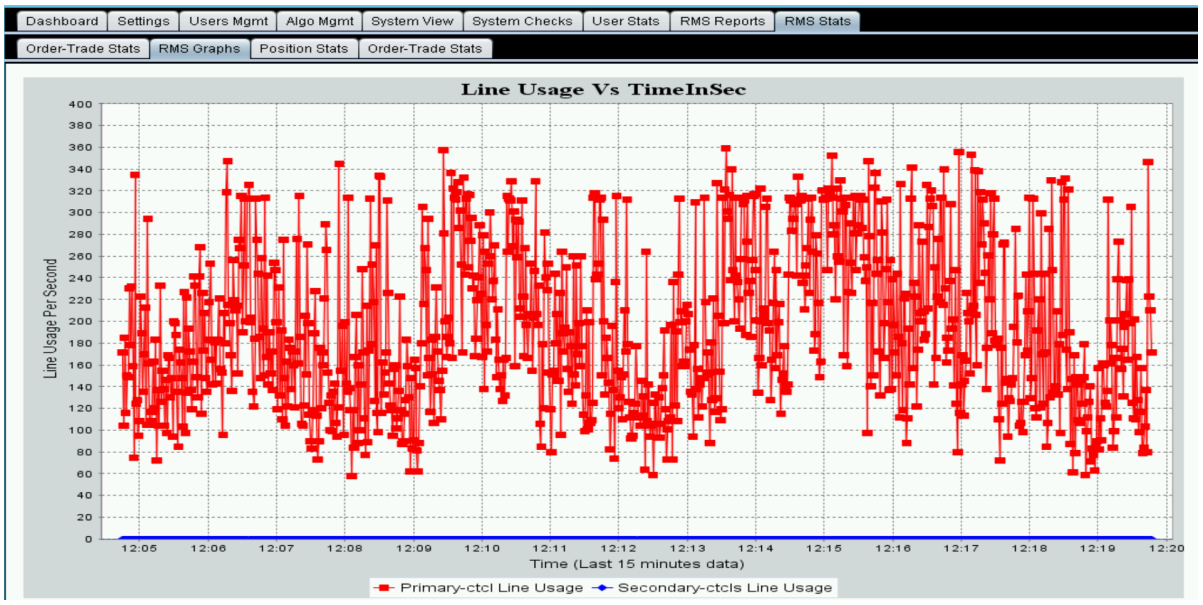


Fig. 9: *RMS* Line Usage Vs Time *Graphs*

User	Pgid	Orders Count	Trades	Order/Trade Ratio	RejectCount	Order rate
user1	300036	4388	2	2194	2	0
user1	300045	0	0	0	0	0
user1	300046	0	0	0	0	0
user1	300047	1339	1	1339	1	0
user1	300041	0	0	0	0	0
user1	300043	14580	1	14580	0	1
user1	300052	39567	12	3297	5	2
user1	300054	6933	0	6933	7	-3
user1	300048	55765	32	1742	24	4
user1	300049	2273	2	1136	2	-5
user1	300050	26274	1	26274	7	2
user1	300051	42114	30	1403	14	1
user6	800006	51899	1	51899	1	5
user6	800007	38651	0	38651	0	-1
user6	800001	131414	3	43804	2	9
user6	800002	14584	0	14584	0	-11
user6	800003	28181	0	28181	0	1
user3	500015	4757	0	4757	0	-1
user3	500014	2867	4	716	1	0
user3	500011	0	0	0	0	0
user3	500023	19071	6	3178	8	1
user3	500022	49432	46	1074	24	3
user3	500021	40030	14	2859	7	0
user3	500020	27042	26	1040	14	-1
user3	500019	39775	26	1529	16	1
user3	500029	7013	22	318	9	-3
user3	500025	14836	31	478	8	0

Fig. 10: *Order Trade Stats*

Appendix:

BOD: Beginning of the day (BOD) consists of activities like transferring contract files and starting the RMS machine.
TBT: Tick By Tick multicast Data received from Exchange.
DPR: Day Price Range is maximum or minimum price of any stock for a complete day defined by exchange.
MWPL: Market Wide Protection Limit.
Stream IP: To join NSE TBT data multicast, NSE gives TBT IP called Stream IP
Stream Port: To join NSE TBT data multicast, NSE gives TBT port called Stream Port
Stream interface: Linux machine interface IP which accepts TBT data.
IA IP: It is Interactive IP address to get connectivity from exchange.
IA Port: It is an Interactive Port required for the Exchange connectivity.
NSE: National Stock Exchange is a place where trades are executed and shares are exchanged.
Partial: Derivative strategy combines legs/derivatives, & when one of the derivatives are not traded while other derivatives intrinsically associated with the strategy are traded, then it is said that the strategy has been executed partially.
Slippage: When one or two derivatives are traded in a manner that the executed sell (buy) price of a derivative strategy is less (more) than the desired or estimated price, then it is called slippage.

FAQs

1.Steps to archive data are as follows:
Archive button -> Login Server -> Refresh system state

2. **Orders Cancelled** should be less than **Orders Confirmed**. Please kill the binary and RMS and reach out support team.

3. Please keep an eye on the no. of **Orders Errored**. All rejected orders consume the message line and give up opportunity of other good orders.

4. If **TBT Messages Accepted** is not increasing even after adding a new position then contract files in RMS must be checked.

5**.** On a **successful BOD,** the value of TBT Messages Accepted should be greater than 10. In other case please contact developer team.