Designed For Diet Odin Trade Automation

Trading Server (Dealing With Trades & Transactions)

Data Transfer To and Fro from Diet Odin To Server.

Diet Odin Application With Market Watch And Buy – Sell Modules.

(Here – The Demo MCX Entry Form As A Simulation Of Diet Odin.)

Data Transfer To Excel Sheet / Data Grid.

Excel Sheet / Data Grid Capable Of Tracking Changes And Generating Buy – Send Signals.

(Here – The Demo Data Grid With Buttons.)

Data Packets Transferred From Excel To OdinBuySellAgent.

Data Packets Transferred From OdinBuySellAgent To Diet Odin.

OdinBuySellAgent (AutoIT Application.)

Receives Input Data From Excel Sheet / Data Grid, Processes It And Transfers It To Diet Odin Application (Here – To The Demo Application.)

Module Made In Auto IT.

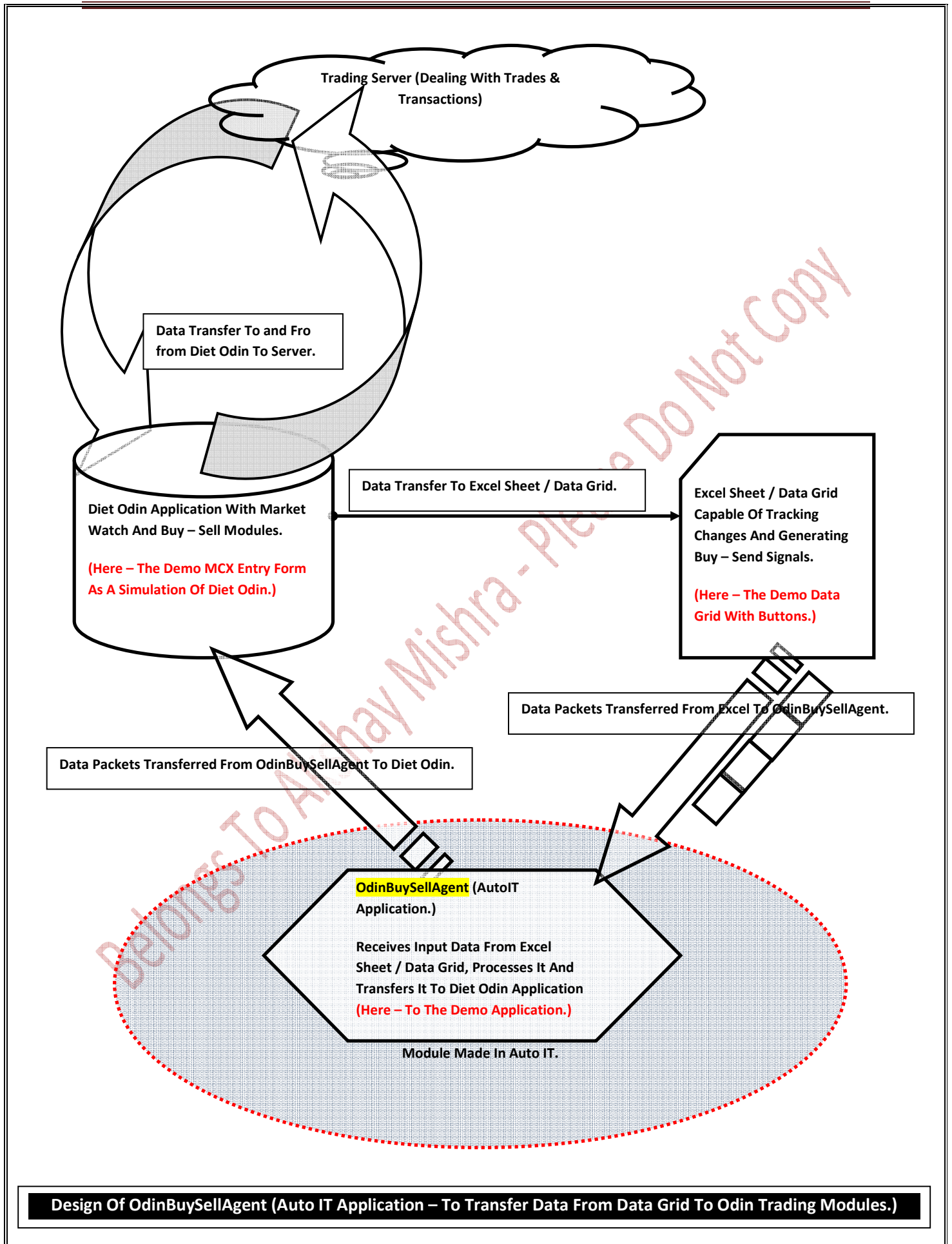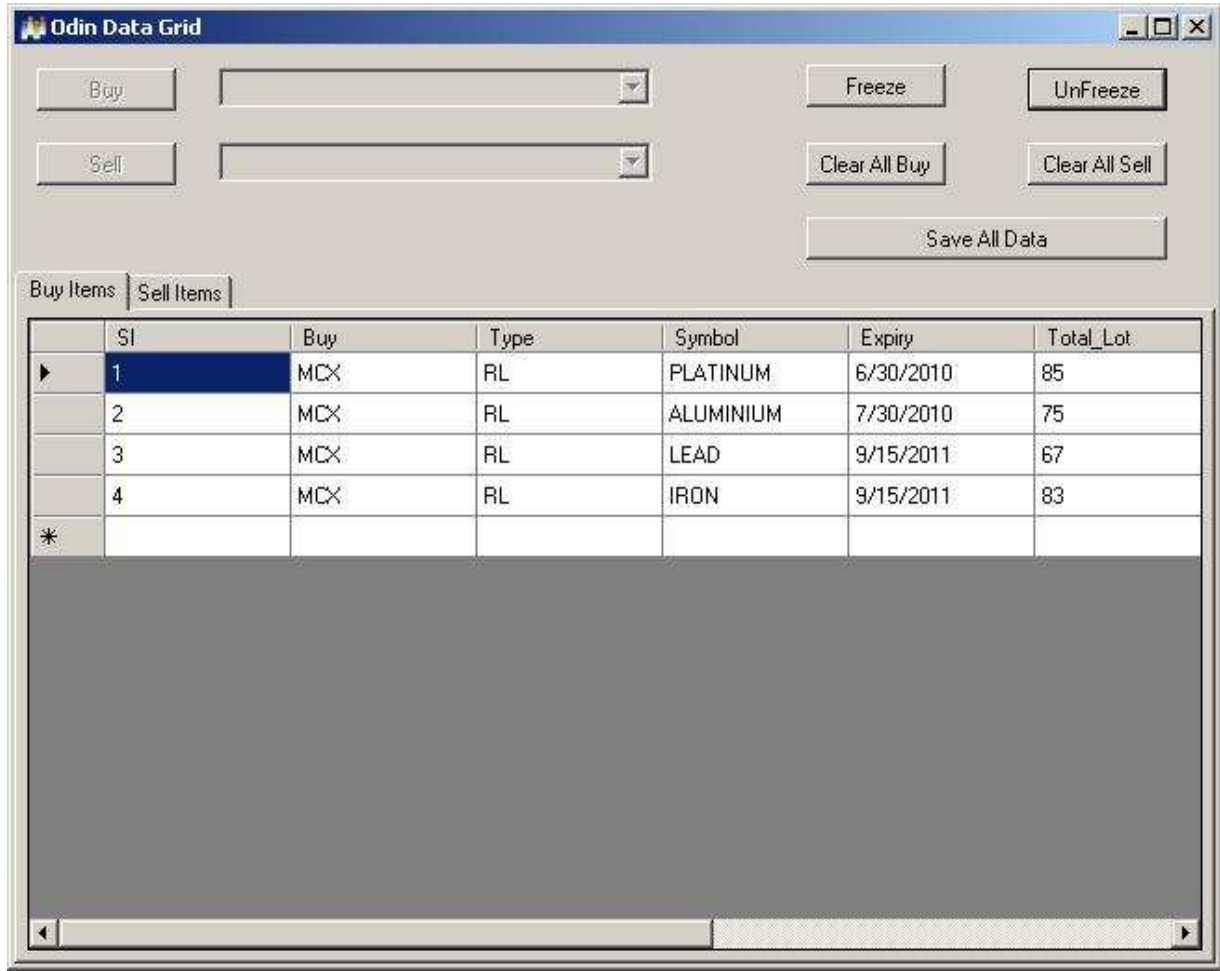**Design Of OdinBuySellAgent (Auto IT Application – To Transfer Data From Data Grid To Odin Trading Modules.)**

User Interface Illustration

Here the database is not frozen. Changing the Transaction Details (cell values) in the middle of any ongoing transaction may lead to corrupted transaction. You can change the cell values and hence "Buy" and "Sell" buttons are grayed out (disabled). Here you can edit the cell values.

Once You "Freeze" the database, you cannot edit the cell (data grid) values anymore – until you "Unfreeze" it back. So the "Buy" and "Sell" buttons will be active only once you Freeze the data. You can select the Buy or Sell Commodity by either the drop down combo box – or by double clicking on the cell values. Selecting a commodity highlights the row in violet color.

You can edit the grid and save the data. The next time you start the application, the saved data shall be reflected in the grid.
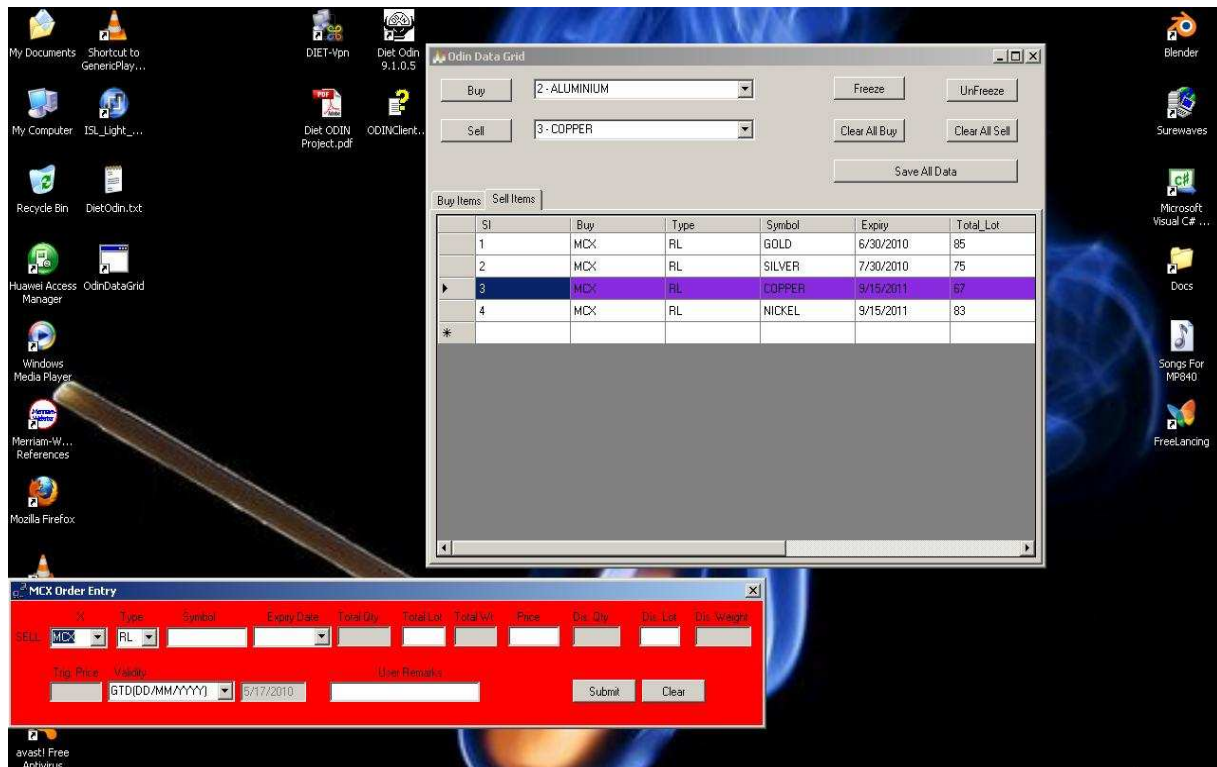
On Freezing the database – the MCX Order Entry window pops up. You can buy or sell the selected commodity on button clicks. Unfreezing the application closes the MCX Order Entry Form.

The AutoIT component is invisible and is triggered (with selected data packets) on button clicks.

The format of data packet is simple – it's a comma separated string of transaction details preceded by a buy / sell command. It is case insensitive (capitals or smalls do not matter).

Example : **"buy,mcx,RL,Gold,20/05/2010,25,18000,5,DAY,MyTradeRemarks"**

The buy/sell commands says (in literal sense) F1 of F2. Other details are the ones to be populated in the MCX Order Entry form. The Excel sheet should have similar data (columns) as the Data grid.

In real time, the excel sheet will pass this data to AutoIT application – which in-turn will interact with the Diet Odin.
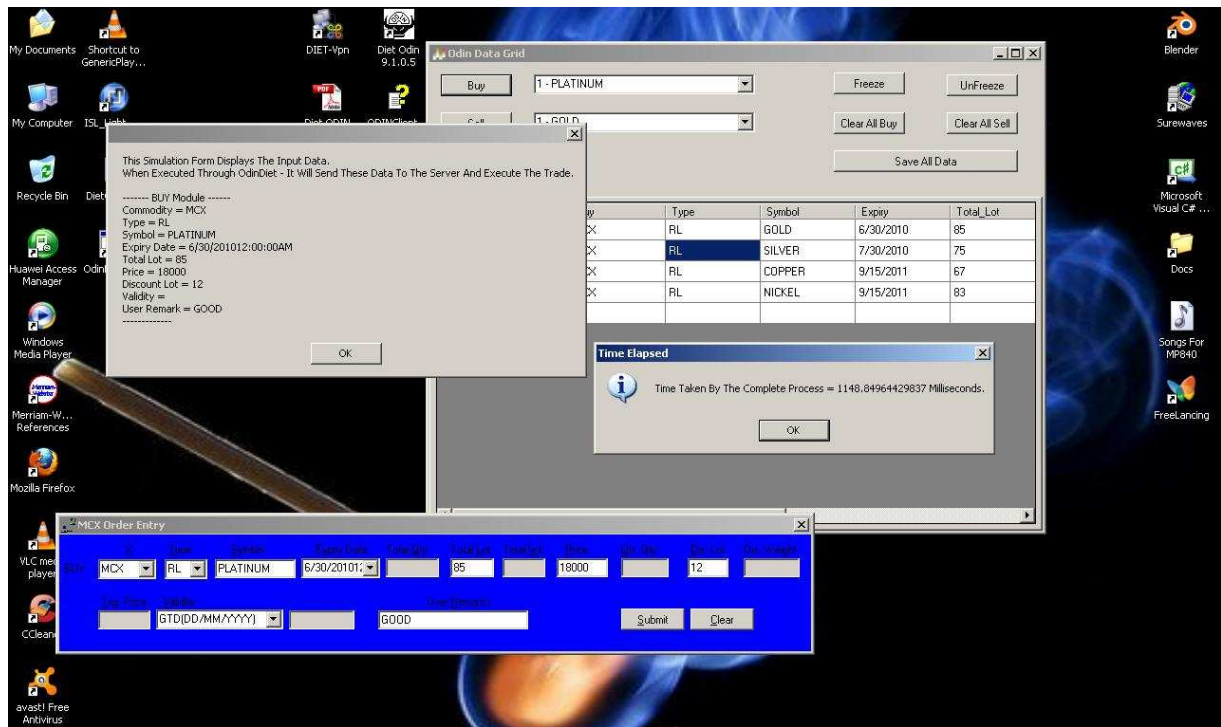
On clicking the submit button, a Pop up window (message box) appears to state the data that was sent from Excel Sheet / Data Grid To AutoIT application – which in turn was send to MCX Order Entry Form and which would be sent to Server if real Diet Odin is used.

Another pop up message box states the time in milliseconds (1/1000th of a second) that was taken from the time AutoIT application received the data packet until it transferred it to MCX Entry Form. This time usually come to be around 370 – 380 ms but depends upon system performance (i.e.: if memory extensive applications like fire fox, music player, etc. are running in parallel, the time taken goes up to 1200 ms = 1.2 seconds per transaction). You can run this application in fast systems (or after closing other applications) and evaluate the time taken per transaction.

In real time scenarios – when we remove this time calculation mechanism (Timer), the speed will be further **enhanced** (as AutoIT application won't have to process time in parallel to inputs).
 You can evaluate the application to your satisfaction and all feedbacks are welcome.