

# uTrade Solutions

## API Version 2.0.1 Client

### Manual

### (Latency Sensitive)



# Table of Contents

---

<b>Table of Contents .....</b>	<b>2</b>
<b>Disclaimer .....</b>	<b>3</b>
<b>1. Company Overview .....</b>	<b>4</b>
<b>2. Preface .....</b>	<b>5</b>
<b>3. Scope of Document.....</b>	<b>5</b>
<b>4. Overview .....</b>	<b>5</b>
<b>5. Steps for writing and running custom strategy .....</b>	<b>6</b>
<b>6. Designing Front-end for Custom Strategy .....</b>	<b>7</b>
<b>7. Steps for writing a strategy (back-end implementation) .....</b>	<b>17</b>
<b>8. Things to do before calling Start Algo .....</b>	<b>20</b>
<b>9. Excel Feeder based Trading .....</b>	<b>21</b>
<b>10. Command Line .....</b>	<b>28</b>
<b>11. Special Notes .....</b>	<b>29</b>
<b>12. Risk Management .....</b>	<b>30</b>
<b>13. Compliance Requirements .....</b>	<b>34</b>
<b>14. System Requirements .....</b>	<b>38</b>
<b>Contact Us .....</b>	<b>39</b>

# Disclaimer

---

This information is intended for sharing the product details for the purpose of potential partnership opportunity. No content and the ideas presented hereby maybe re-used, re-distributed or discussed outside of the organization where it is presented (without the prior consent of the author of this document). None of the ideas for financial trading technology presented in these slides maybe copied or shared with parties that could be potentially competing against uTrade Solutions. The information shall not be distributed or used by any person or entity in any jurisdiction or countries where such distribution or use would be contrary to the applicable laws or Regulations. It is advised that prior to acting upon this information, independent consultation / advise may be obtained and necessary due diligence, investigation etc. may be done at the users end. You may also contact us directly for any questions or clarifications. All statements regarding the future are subject to inherent risks and uncertainties, and many factors may lead to actual profits or losses & developments deviating substantially from what has been expressed or implied in such statements.

# 1. Company Overview

---

**uTrade Solutions Pvt. Ltd. is a capital markets trading technology company providing open source-based multi-asset trading platform, algorithms and analytics for global financial institutions including brokers, banks, exchanges, other financial institutions and their end customers globally.**

At uTrade, we provide software products having a strong focus on user experience, leverage of automated technologies enabling financial institutions and their users to trade and analyze information more efficiently.

Our product suite includes the following:

- ✓ **Trading platform (Retail and Institutional)** – with full suite application and web-based front ends (Including admin functions, risk management, order management, connectivity to exchanges etc.)
- ✓ **Low latency algorithmic trading platform** – Used in exchange co-location environment or in client data centres/cloud for fastest access to markets to execute arbitrage, market making, execution, quant based and various other strategies.
- ✓ **Analytics** – platform that supports portfolio-level, pre-trade and post-trade analytics, content creation, management and distribution.
- ✓ **Custom Solutions** – We also customise various trading technology and analytics products, including modules of our own products, to improve the trading workflow of our partner users.

We have built our products from ground-up with a modular architecture in order to effectively address current and rapidly evolving user needs. We have also filed for 6 patents in India and 1 patent in US/UK to lead innovation in the trading life cycle.

**uTrade Solutions was awarded as most innovative firm by Fintech Innovation Labs UK, and Confederation of Indian Industries, as well as a leading start-up by NASSCOM, India.**

## 2. Preface

---

The objective of this document is to guide the users to understand uTrade's latency sensitive API interface for coding custom strategies. This document explains all the features available in the API 2.0.1

## 3. Scope of Document

---

The scope of this document is to specify the steps to write custom strategy (both front-end and back-end logic) which can work with uTrade's latency sensitive algorithmic trading platform. It explains various features of the API 2.0.1, does not cover the in-built strategies which come with the uTrade platform.

## 4. Overview

---

- ✓ uTrade's API 2.0.1 provides backend infrastructure, which allows you to create your own latency sensitive strategies and run with uTrade's latency sensitive trading platform.
- ✓ Custom strategies can be written only in C++ (Linux platform)
- ✓ New custom strategies can be integrated and run in the platform on the fly with no effect on the already running strategies. They can be run from the same front-end.
- ✓ Risk Management features are also available, same as the trading platform.
- ✓ You will have to code only for backend strategy.
- ✓ You need to generate Frontend using an in-built tool.
- ✓ Custom strategy can be run with both TBT and snapshot feed. Latest data (order book) is provided to the strategy on request-basis.
- ✓ API 2.0.1 supports all segments of NSE, BSE, SGX, CFH

## 5. Steps for writing and running custom strategy

---

- ✓ You need to generate front-end using an in-built tool
- ✓ You need to code and create custom strategy .so file (as shared object or dynamic library)
- ✓ At back-end, you need to copy the .so to custom strategy folder.
- ✓ Now, to run the strategy from front-end, you need to specify the name of .so file.
- ✓ Automatically, the strategy gets loaded in the application
- ✓ Now you can run the strategy

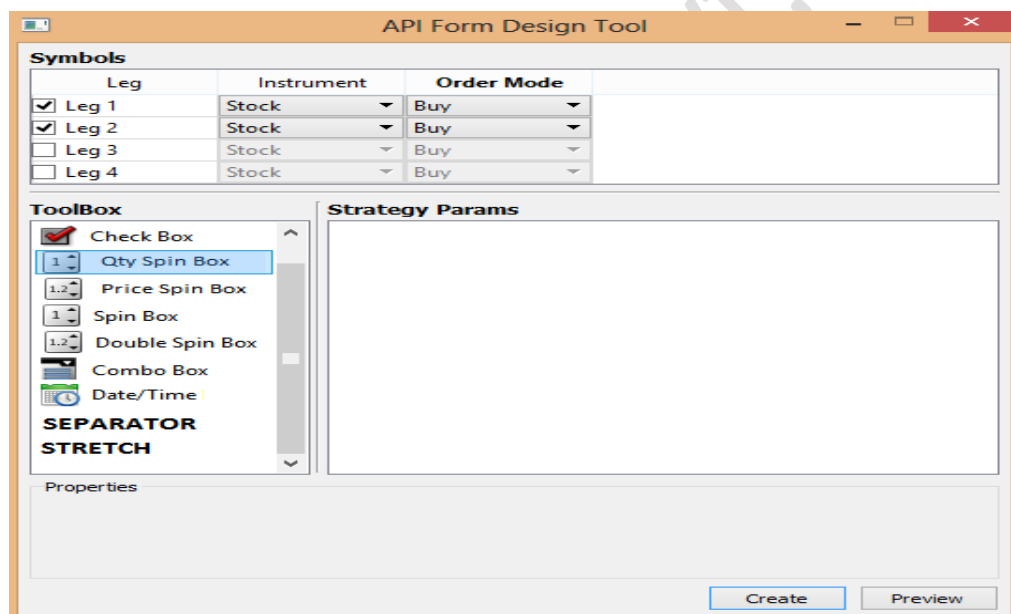
uTrade Solutions Pvt. Ltd.

## 6. Designing Front-end for Custom Strategy

Select parameters from tool box that you wish to see on the front-end window.

Depending on the type of parameter, front-end form gets generated where you can fill values based on logic, and integrate with the custom strategy back-end implementation.

### 6.1 Tool Box



Select number of legs you want to view on the front-end to run your custom strategy. For e.g. if you want to create a 2 leg strategy, select two legs.

The screenshot shows the 'API Form Design Tool' window. The 'Symbols' table has the following data:

Leg	Instrument	Order Mode
<input checked="" type="checkbox"/> Leg 1	Stock	Buy
<input checked="" type="checkbox"/> Leg 2	Stock	Buy
<input type="checkbox"/> Leg 3	Future	Buy
<input type="checkbox"/> Leg 4	Option	Buy

The 'Order Mode' dropdown for Leg 2 is open, showing 'Buy' as the selected option. The 'ToolBox' on the left contains various UI controls like Radio Button, Check Box, Qty Spin Box, Price Spin Box, Spin Box, Double Spin Box, Combo Box, and Date/Time. The 'Strategy Params' section is empty. At the bottom right are 'Create' and 'Preview' buttons.

Select default instrument for different legs. If you want to run cash future strategy, select stock for first leg and future for second leg.

The screenshot shows the 'API Form Design Tool' window. The 'Symbols' table has the following data:

Leg	Instrument	Order Mode
<input checked="" type="checkbox"/> Leg 1	Stock	Buy
<input checked="" type="checkbox"/> Leg 2	Stock	Buy
<input type="checkbox"/> Leg 3	Stock	Buy
<input type="checkbox"/> Leg 4	Stock	Buy

The 'Order Mode' dropdown for Leg 1 is open, showing 'Buy' as the selected option. The 'ToolBox' on the left contains various UI controls like Radio Button, Check Box, Qty Spin Box, Price Spin Box, Spin Box, Double Spin Box, Combo Box, and Date/Time. The 'Strategy Params' section is empty. At the bottom right are 'Create' and 'Preview' buttons.

Select, buy or sell options for different legs. For e.g. if you want to sell cash and buy futures, select sell for leg one and buy for second leg.



The screenshot shows the 'API Form Design Tool' window. The 'Symbols' table at the top lists four legs: Leg 1 (checked, Stock, Sell), Leg 2 (checked, Future, Buy), Leg 3 (unchecked, Stock, Buy), and Leg 4 (unchecked, Stock, Buy). The 'ToolBox' on the left has 'Qty Spin Box' selected. The 'Strategy Params' area on the right contains two 'Qty Spin Box' entries. The 'Properties' section at the bottom shows the label 'quantity\_1', Min Val '0', Max Val '10000', and Def Val '100'. 'Create' and 'Preview' buttons are at the bottom right.

Leg	Instrument	Order Mode
<input checked="" type="checkbox"/> Leg 1	Stock	Sell
<input checked="" type="checkbox"/> Leg 2	Future	Buy
<input type="checkbox"/> Leg 3	Stock	Buy
<input type="checkbox"/> Leg 4	Stock	Buy

**ToolBox**

- Radio Button
- Check Box
- Qty Spin Box**
- Price Spin Box
- Spin Box
- Double Spin Box
- Combo Box
- Date/Time
- SEPARATOR

**Strategy Params**

- Qty Spin Box
- Qty Spin Box

**Properties**

Label: quantity\_1    Min Val: 0    Max Val: 10000    Def Val: 100    Update

Create    Preview

Select Spin box for entering quantity or whole values. Define label like Order Quantity, define value range min and max, enter default value you want to trade on and update.

The screenshot shows the 'API Form Design Tool' window. The 'Symbols' table and 'ToolBox' are the same as in the previous screenshot, but 'Price Spin Box' is now selected in the 'ToolBox'. The 'Strategy Params' area on the right contains two 'Qty Spin Box' entries and one 'Price Spin Box' entry. The 'Properties' section at the bottom shows the label 'price\_2', Min Val '0.00', Max Val '99.00', and Def Val '0.00'. 'Create' and 'Preview' buttons are at the bottom right.

Leg	Instrument	Order Mode
<input checked="" type="checkbox"/> Leg 1	Stock	Sell
<input checked="" type="checkbox"/> Leg 2	Future	Buy
<input type="checkbox"/> Leg 3	Stock	Buy
<input type="checkbox"/> Leg 4	Stock	Buy

**ToolBox**

- Radio Button
- Check Box
- Qty Spin Box
- Price Spin Box**
- Spin Box
- Double Spin Box
- Combo Box
- Date/Time
- SEPARATOR

**Strategy Params**

- Qty Spin Box
- Qty Spin Box
- Price Spin Box

**Properties**

Label: price\_2    Min Val: 0.00    Max Val: 99.00    Def Val: 0.00    Update

Create    Preview

Select double spin box for entering price or rational numbers. Define label like Min Profit, define price range min and max, set default price you want to trade on and update.

The screenshot shows the 'API Form Design Tool' window. The 'Symbols' section contains a table with columns 'Leg', 'Instrument', and 'Order Mode'. The 'ToolBox' on the left lists various UI components, with 'Combo Box' selected. The 'Strategy Params' section on the right shows a list of parameters: 'Qty Spin Box', 'Qty Spin Box', 'Price Spin Box', and 'Combo Box'. The 'Properties' section at the bottom shows the 'Label' as 'combobox\_3' and the 'Value' as an empty field. Buttons for 'Add', 'Remove', 'Update', 'Create', and 'Preview' are visible.

Leg	Instrument	Order Mode
<input checked="" type="checkbox"/> Leg 1	Stock	Sell
<input checked="" type="checkbox"/> Leg 2	Future	Buy
<input type="checkbox"/> Leg 3	Stock	Buy
<input type="checkbox"/> Leg 4	Stock	Buy

**ToolBox**

- Radio Button
- Check Box
- Qty Spin Box
- Price Spin Box
- Spin Box
- Double Spin Box
- Combo Box
- Date/Time

**Strategy Params**

- Qty Spin Box
- Qty Spin Box
- Price Spin Box
- Combo Box

**Properties**

Label: combobox\_3    Value:    Add    combobox\_3    Remove    Update

Create    Preview

Combo will give a drop-down option. Label the Combo like Hedge Methods, add values like best bid, best ask, etc. you can add or remove the values, click on the value you want to see as default and update.

The screenshot shows the 'API Form Design Tool' window. The 'Symbols' section is the same as in the previous image. The 'ToolBox' on the left has 'Date/Time' selected. The 'Strategy Params' section on the right shows a list of parameters: 'Qty Spin Box', 'Qty Spin Box', 'Price Spin Box', 'Combo Box', and 'Date/Time'. The 'Properties' section at the bottom shows the 'Label' as 'datetime\_4' and the 'Value' as an empty field. Buttons for 'Update', 'Create', and 'Preview' are visible.

Leg	Instrument	Order Mode
<input checked="" type="checkbox"/> Leg 1	Stock	Sell
<input checked="" type="checkbox"/> Leg 2	Future	Buy
<input type="checkbox"/> Leg 3	Stock	Buy
<input type="checkbox"/> Leg 4	Stock	Buy

**ToolBox**

- Radio Button
- Check Box
- Qty Spin Box
- Price Spin Box
- Spin Box
- Double Spin Box
- Combo Box
- Date/Time

**Strategy Params**

- Qty Spin Box
- Qty Spin Box
- Price Spin Box
- Combo Box
- Date/Time

**Properties**

Label: datetime\_4    Value:    Update

Create    Preview

Separator adds a horizontal line and date/time adds a box with timer. You only have the option to edit the label.

The screenshot shows the 'API Form Design Tool' window. The 'Symbols' section contains a table with four legs. The 'ToolBox' on the left lists various UI components, with 'Radio Button' selected. The 'Strategy Params' section on the right shows a list of parameters, with 'Radio Button' highlighted. The 'Properties' section at the bottom shows the 'Label' as 'radio\_5' and the 'Value' as an empty field. Buttons for 'Add', 'Remove', 'Update', 'Create', and 'Preview' are visible.

Leg	Instrument	Order Mode
<input checked="" type="checkbox"/> Leg 1	Stock	Sell
<input checked="" type="checkbox"/> Leg 2	Future	Buy
<input type="checkbox"/> Leg 3	Stock	Buy
<input type="checkbox"/> Leg 4	Stock	Buy

**ToolBox**

- ☒ Radio Button
- ☒ Check Box
- Qty Spin Box
- Price Spin Box
- Spin Box
- Double Spin Box
- Combo Box
- Date/Time
- SEPARATOR
- STRETCH

**Strategy Params**

- Qty Spin Box
- Qty Spin Box
- Price Spin Box
- Combo Box
- SEPARATOR
- Date/Time
- Radio Button**

**Properties**

Label: radio\_5    Value:    Add    radio\_5    Remove    Update

Create    Preview

Radio allows you to select either of the options; you have to add minimum 2 values in radio. Label will not appear on the screen, you can only view the values, select one of the value and update.

The screenshot shows the 'API Form Design Tool' window. The 'Symbols' section contains a table with four legs. The 'ToolBox' on the left lists various UI components, with 'Check Box' selected. The 'Strategy Params' section on the right shows a list of parameters, with 'Check Box' highlighted. The 'Properties' section at the bottom shows the 'Label' as 'checkboxbox\_6' and the 'Default State' as an unchecked box. Buttons for 'Update', 'Create', and 'Preview' are visible.

Leg	Instrument	Order Mode
<input checked="" type="checkbox"/> Leg 1	Stock	Sell
<input checked="" type="checkbox"/> Leg 2	Future	Buy
<input type="checkbox"/> Leg 3	Stock	Buy
<input type="checkbox"/> Leg 4	Stock	Buy

**ToolBox**

- ☒ Radio Button
- ☒ Check Box**
- Qty Spin Box
- Price Spin Box
- Spin Box
- Double Spin Box
- Combo Box
- Date/Time
- SEPARATOR
- STRETCH

**Strategy Params**

- Qty Spin Box
- Qty Spin Box
- Price Spin Box
- Combo Box
- SEPARATOR
- Date/Time
- Radio Button
- Check Box**

**Properties**

Label: checkbox\_6    ☐ Default State    Update

Create    Preview

Stretch will create a gap between two different parameters. Check box allows you to select multiple options; you can select more than one option. Add label to the check box, select on default state if you want this particular box to be checked.

The **API Form Design Tool** window is used to configure the front-end interface. It includes the following sections:

- Symbols:** A table for configuring multiple legs of an order.
 

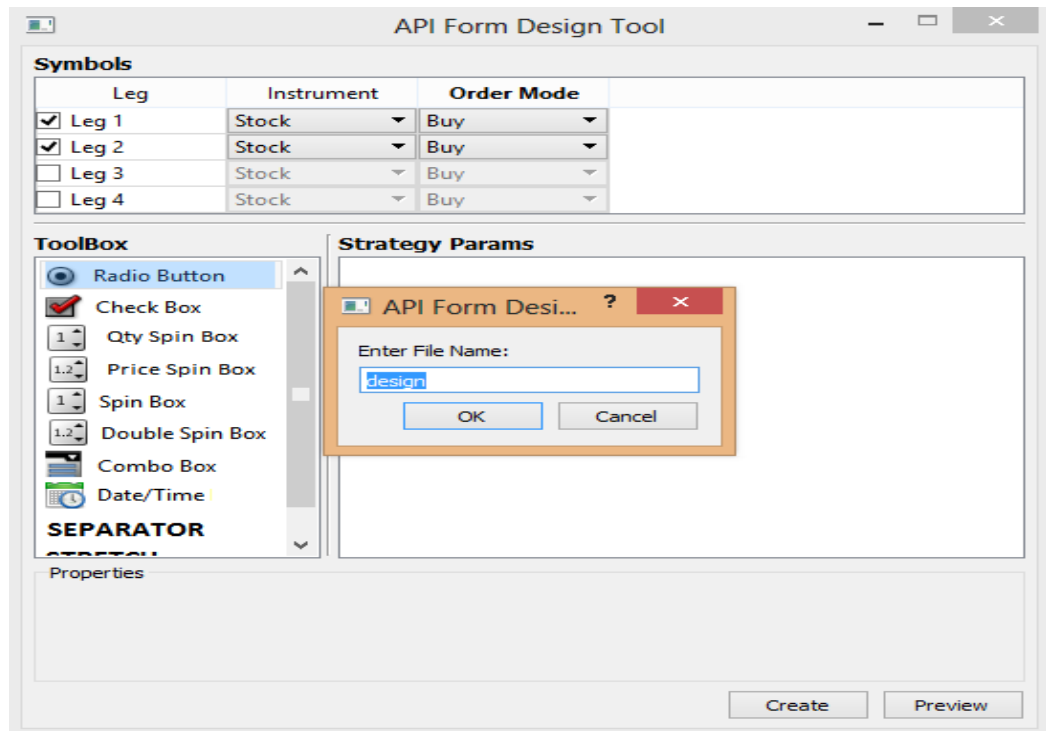
Leg	Instrument	Order Mode
<input checked="" type="checkbox"/> Leg 1	Future	Buy
<input checked="" type="checkbox"/> Leg 2	Stock	Sell
<input type="checkbox"/> Leg 3	Stock	Buy
<input type="checkbox"/> Leg 4	Stock	Buy
- ToolBox:** A list of UI components available for design, including Radio Button, Check Box, Qty Spin Box, Price Spin Box, Spin Box, Double Spin Box, Combo Box, Date/Time, SEPARATOR, and STRETCH.
- Strategy Params:** A list of parameters to be included in the strategy, such as Qty Spin Box, Price Spin Box, Combo Box, Date/Time, Radio Button, Check Box, and STRETCH.
- Properties:** Fields for configuring the selected component. For example, for a 'Total Qty' label, it shows Min Val (0), Max Val (10000), and Def Val (100). An 'Update' button is present.
- Buttons:** 'Create' and 'Preview' buttons at the bottom right.

Click on preview to see the window which will appear on front-end.

The **Preview** window displays the final front-end trading interface based on the configuration in the design tool. It includes the following sections:

- Order Entry:** Fields for Mode (SELL/BUY), Market (NSE), Instrument (Stock/Future), Symbol, Series, Expiry, Opt Mode, Strike, and Lot Size.
- Strategy Params:** Fields for Total Qty (100), Order Qty (10), Min Profit (100.00), Hedge Methods (Best Bid/Ask), Timer (12:11:57.723), and checkboxes for Product, Actual, Chk Spread, and Bid Diff.
- Buttons:** 'New', 'Edit', 'Commit', 'Add', 'Remove', 'Run', 'Stop', and 'Stop All' buttons.
- Table:** A table to display order details with columns: Id, State, Comment, Risk Status, SYMBOL LEG1, Order Mode 1, SYMBOL LEG2, Order Mode 2, Total Qty, and Order.

Based on your criteria and your default selection, this is how your front-end window will appear. If you don't want to modify, click on create and add enter file name.



The file will be created, go to your application folder and find the name of the file, open in notepad to view the code.

The **code** thus generated will be:

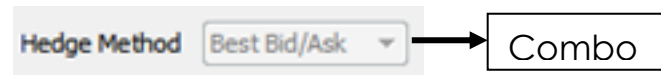
```
[SYMBOL]
SYMBOL LEG1=UINT64:F
Order Mode 1=UCHAR:1
SYMBOL LEG2=UINT64:F
Order Mode 2=UCHAR:0
[STRATEGY_PARAMS]
Total Qty=UINT64:SPINBOX:0:99999:1:Q
Order Qty=UINT64:SPINBOX:0:99999:1:Q
Min Profit=FLOAT:DSPINBOX:0.00:999.00:0.05:P
Hedge Method=COMBO:Market Order:Best Bid/Ask:Best
Ask/Bid:Best Bid/Ask
SEPARATOR=1
Timer=TIMER
Spd Type=RADIO:Product:Actual:Def Rule:Actual
STRETCH=1
Bid Diff Chk=BOOL:0
Bid Diff Val=UINT64:SPINBOX:0:999:0
[OTHER]
```

This file is required for **back-end implementation**.

## 6.2 How to use different parameters

### 6.2.1 COMBO

Name=COMBO:Data 1:Data 2:Data 3

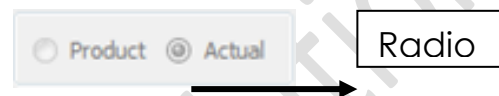


Combo is used to create any parameter with drop down option, like in above mentioned image you can click on drop down option and select different hedge methods.

### 6.2.2 RADIO

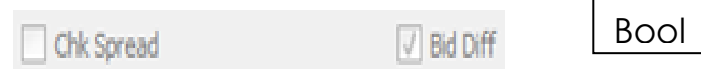
Name=RADIO:Radio 1:radio 2:radio 3:radio 2

Last value (radio 2) is default checked radio button.



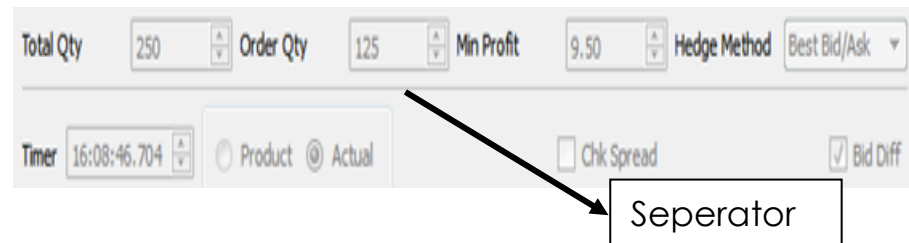
Radio is used where parameter can be selected among multiple options; however you can choose only one option. Radio is represented by circle. In above image you have multiple options, but you can only select either of the two.

### 6.2.3 CHECK BOX



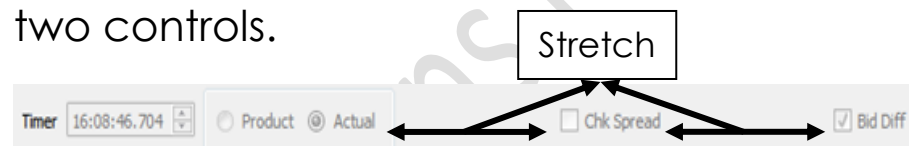
Checkbox is used where parameter can be selected as true/false. Bool is represented by a check/ square box.

## 6.2.4 SEPARATOR will create a horizontal line



Separator is used where you want to divide the window to show different parameters in different row. In above image total qty, order qty, min profit and hedge methods are separated by horizontal line from other parameters like timer, product, actual, chk spread and bid diff.

## 6.2.5 STRETCH will create some space in between two controls.

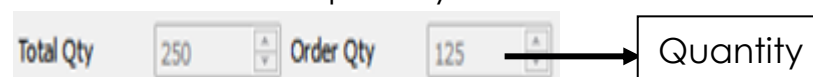


Stretch is used to create gap between the two parameters. In above image you can see there is a stretch between actual and chk spread, and chk spread and bid diff.

## 6.2.6 Create box for quantity parameter

**Qty=UINT64:Q**

Q will be used for quantity.

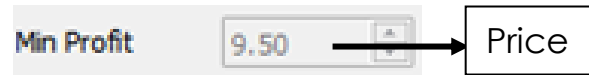


To create box for values in whole number UINT64 is used. In above image to define quantity, which is a whole number, UINT64 is used. (SPIN BOX)

## 6.2.7 Create box for price parameter

`Min Profit=FLOAT:P`

P will be used for Price.



To create box for values in natural number FLOAT is used. In above image to define price, which is a natural number, FLOAT is used. (DOUBLE SPIN BOX)

Note:

1. Whenever you are creating a symbol bar, order mode with it is compulsory. e.g.

`SYMBOL LEG1=UINT64`

`Order Mode 1=UCHAR`



2. Line starting with # is comment and can't be used. e.g.

`#SYMBOL LEG3=UINT64`



## 7. Steps for writing a strategy (back-end implementation)

### 7.1 Create an external interface

Your strategy will run in the same process as muTrade back-end application, your strategy will be created as a shared object.

The following functions are mandatory to be present:

```
extern "C"
```

```
{
```

```
// inside extern "C" Specifier
```

```
std :: string getFrontendDesign();
```

It is used to get the script file which will be passed to frontend, further it will render the GUI design using which you strategy will run.

```
void * getDriver(void * params);
```

It will be called when front-end issues strategy run command to the back-end.

Here params will point to object of class SG::Parameters defined in sgLib/sgApiParameters.h

```
}
```

Both of the above functions will need to be place under <extern "C"> specifier.

The strategy will be running inside the HFT back-end.

See [API-Documentation](#)

## 7.2 Inherit from API2::SGContext

Here you need to write your own class which inherits [API2::SGContext](#).

[API2::SGContext](#) provides various functionalities such as Market Data Subscription, Order Execution, OrderBook and Net Position processing and will provide your derived class various Event call-backs for different types of Order Confirmations.

See [API-Documentation](#)

## 7.3 Write your core strategy logic in your derived class.

Here you need to write your core strategy logic, which will be executed on various Events provided such as

### 7.3.1 Market Data Event:

For enabling this event use:  
marketDataEventRequired = true while starting the algo

See [API-Documentation](#)

This event call-back is generated whenever there is a change in market feed for the subscribed Feed Type (TBT Feed or Snapshot Feed)

### 7.3.2 On Frontend Command:

This event call-back is generated whenever there is a command from frontend, like stop/modify strategy etc

See [API-Documentation](#)

### 7.3.3 On Order Confirmations:

This event call-back is generated whenever an Exchange Order Confirmation (new/modify confirmed/rejected/partially-filled/filled) is received. The call-back is generated after process the Confirmation.

See [API-Documentation](#)

### 7.3.4 General Event call-back:

([API2::SGContext::onDefaultEvent](#))

This call-back is generated if none out of the above event has occurred.

For enabling this event use `marketDataEventRequired = false` while starting the algo using [API2::SGContext::reqStartAlgo\(\)](#)

See [API-Documentation](#)

## 7.4 Drive the algo:

Read the front-end parameters and construct object for your derived class and call the [API2::SGContext::reqStartAlgo\(\)](#)

## 8. Things to do before calling Start Algo

---

### 8.1 Add all the instruments to be used in the strategy.

- To add an instrument use: **API2::SGContext** or **API2::SGContext::createNewInstrument**
- If required for market data subscription, set the subscription parameters accordingly.

See [API-Documentation](#). By calling the above functions, a new instrument is created and market data is subscribed as requested, for that particular symbol (configurable). This function will set up instrument, which will aid you with functionalities such as to monitor, Net Positions, Order Status etc.

### 8.2 Choose start Algo function argument

After market data subscription, you can call **API2::SGContext::reqStartAlgo** function; this takes a single argument, **marketDataEventRequired**.

- **If it is set to true**, then strategy will receive an event, every time feed changes in any symbol, registered.  
All symbols that are subscribed infra will monitor them and whenever feed change is detected event callback

**onMarketDataEvent** is generated

- **If it is set to false**, then strategy is responsible to update quote, to receive latest quotes.

Whenever a general event call-back **API2::SGContext::onDefaultEvent** is generated infinitely. The algo is required to update Market Data, by calling **API2::SGContext::reqUpdateMarketData()** to update market data.

## 9. Excel Feeder based Trading

Excel feeder is an api based algo trading application software, which is not used for latency sensitive algo trading. You can create your own strategies and implement them.

Login to your account using muTrade application; fill in your login credentials.

**Admin - Login**

Enter your User Id

User Id: 18

Password: ●●●●●●●●

☒ Archive Data ☐ Replay Trades Only

☐ Do Not Replay ☐ Update Scrip Detail

After successful login, go to the main menu bar and click on **"Orders"**, select **"Excel Order"**.

The screenshot shows the muTrade application interface. The 'Orders' menu is open, displaying various trading strategies and their keyboard shortcuts. The 'Excel Order' option is highlighted. The background shows the main application window with a menu bar and a data table.

SERIES	EXPIRY	OPT TYPE	STRIKE PRICE
1512	INDHOT...	FQ	
1594	INFY EQ	EQ	

Excel order will open a new window (following is the image)

**uTrade Excel Feeder**

☐ Multicast    Host: 233.1.1.13    Port: 43210    Interface: 10.0.0.140

☒ Broadcast    Port: 20014

**Select Columns**

- ☒ Security Id
- ☒ Market
- ☒ Scrip Name
- ☒ Bid Qty
- ☒ Bid Price
- ☒ Ask Price
- ☒ Ask Qty
- ☒ Last Trade Qty
- ☒ Last Trade Price
- ☒ Trade Value
- ☒ Volume
- ☒ Open

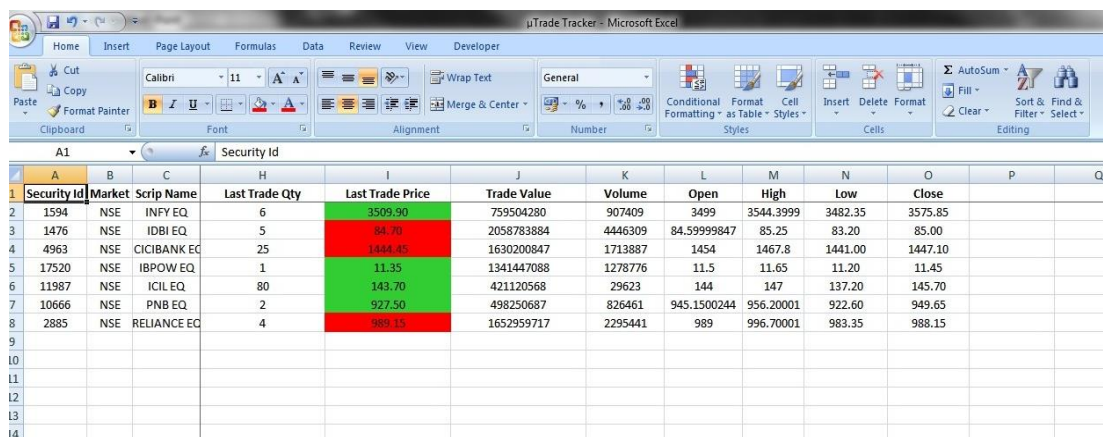
Select the option (Multicast or Broadcast) from where you want to get the feeds. You can also select the columns you want to display on your excel file.

You can also change the location of various columns.

Once you have selected the columns click on **“Order”**. It will open muTrade Application tracker. There are 5 sheets in muTrade Tracker namely:

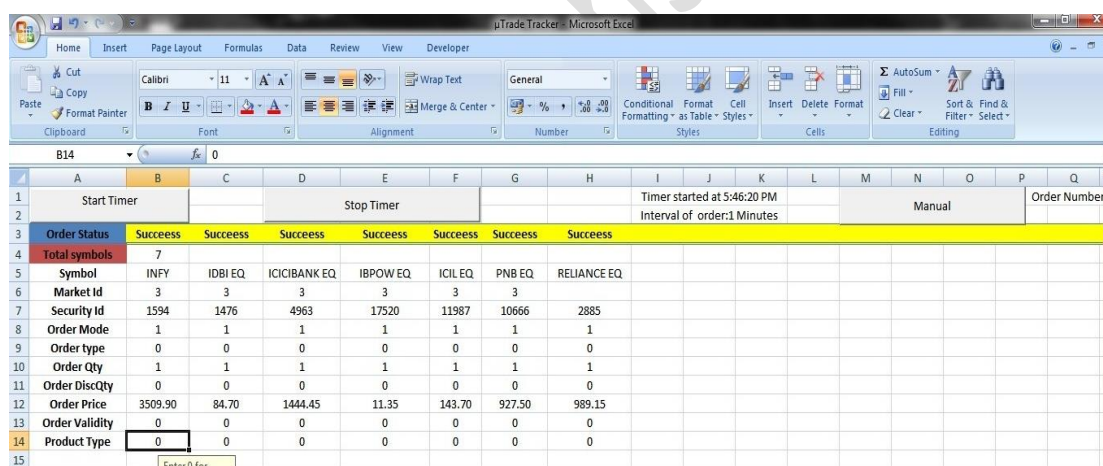
1. Feed
2. Order
3. LTP
4. Log
5. OHLC

**Feed:** Its similar to Market watch, you simply need to add Security ID, Market and the Scrip Name. You will receive the feeds automatically.



Security Id	Market	Scrip Name	Last Trade Qty	Last Trade Price	Trade Value	Volume	Open	High	Low	Close
1594	NSE	INFY EQ	6	3509.90	759504280	907409	3499	3544.3999	3482.35	3575.85
1476	NSE	IDBI EQ	5	84.70	2058783884	4446309	84.59999847	85.25	83.20	85.00
4963	NSE	CICIBANK EQ	25	1444.45	1630200847	1713887	1454	1467.8	1441.00	1447.10
17520	NSE	IBPOW EQ	1	11.35	1341447088	1278776	11.5	11.65	11.20	11.45
11987	NSE	ICIL EQ	80	143.70	421120568	29623	144	147	137.20	145.70
10666	NSE	PNB EQ	2	927.50	498250687	826461	945.1500244	956.20001	922.60	949.65
2885	NSE	RELIANCE EQ	4	989.15	1652959717	2295441	989	996.70001	983.35	988.15

**Order:** It allows you to place orders manually or in an automated manner after a certain time interval as defined by the trader.



Order Status	Success	Success	Success	Success	Success	Success
Total Symbols	7					
Symbol	INFY	IDBI EQ	CICIBANK EQ	IBPOW EQ	ICIL EQ	PNB EQ
Market Id	3	3	3	3	3	3
Security Id	1594	1476	4963	17520	11987	10666
Order Mode	1	1	1	1	1	1
Order type	0	0	0	0	0	0
Order Qty	1	1	1	1	1	1
Order DiscQty	0	0	0	0	0	0
Order Price	3509.90	84.70	1444.45	11.35	143.70	927.50
Order Validity	0	0	0	0	0	0
Product Type	0	0	0	0	0	0

If you chose to place order in an automated manner, you need to add a time limit after which the order will be generated automatically and start the timer, if you wish to stop the orders from getting generated you may stop the timer.

You can also place orders manually.

**Order Status** gives an update whether the order placed is successful or if there is something lacking or some information left unfilled.

**Total Symbols:** allow you to define the number of orders you want to place, for e.g. if you want to buy first three instruments then you'll type 3.

In case you want to buy last two symbols, then you need to rearrange the symbols and type 2 so that first two instruments order gets generated.

**Symbol:** Allows you to enter symbols you want to trade on.

**Market ID:** Select and Enter Market ID. View comments when you click on Market ID.

**Security ID:** Enter the Security ID of each scrip

**Order Mode:** Select buy/sell order. Click on the order mode to read comments.

**Order Type:** Allows you to select limit/market order. Click on the order type to view comments.

**Order Quantity:** Define quantity you wish to buy for each symbol.

**Order Disclose Quantity:** Enter disclose quantity of each symbol as defined by the exchange.

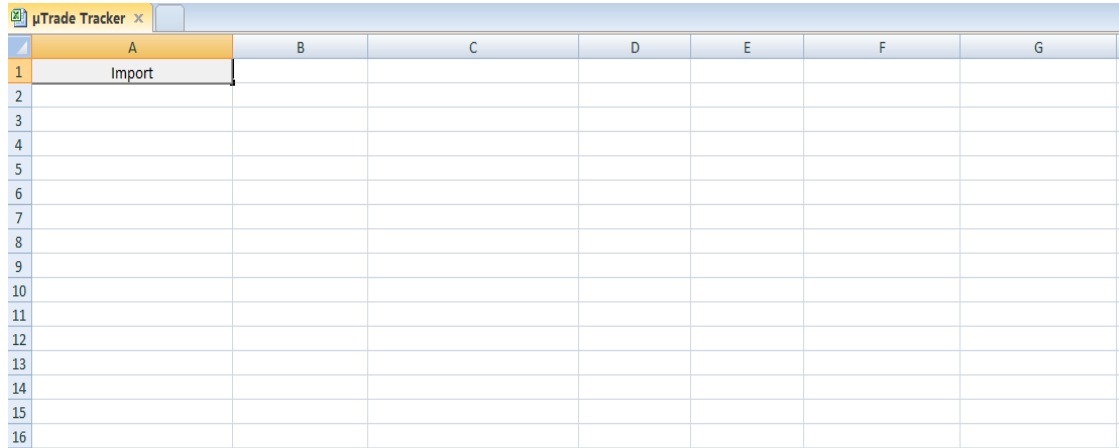
**Order Price:** It is the price that the trader will provide by applying his strategy. This row will be mapped to the strategy page where the order price gets generated.

**Order Validity:** Select Day/IOC order. Click on the order validity to view comments.

**Product type:** Select Intraday/delivery order. Click on the product type to view comments.



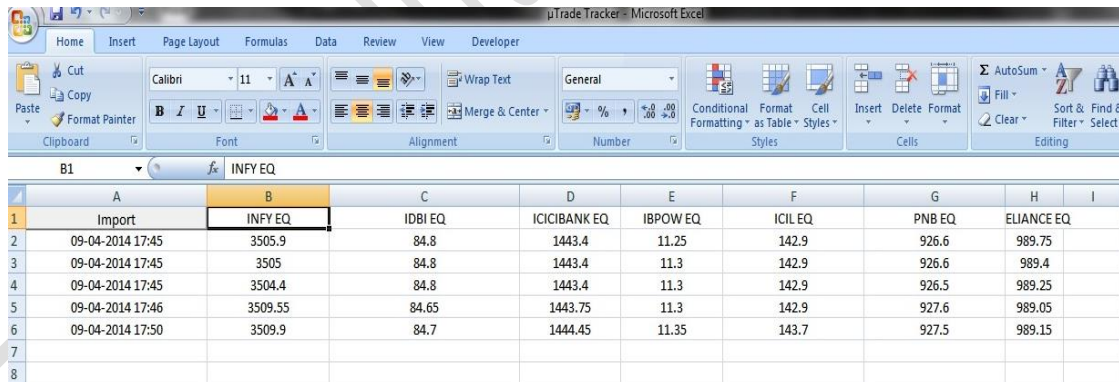
**LTP:** LTP gets auto populated when you click on Import Button. You will see all the symbols added to your orders page getting populated to LTP sheet.



	A	B	C	D	E	F	G
1	Import						
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							

In case of automated trading, the LTP will get updated after specific time interval as defined by the trader.

In case of manual trading, the LTP gets updated every time the trader places an order.



	A	B	C	D	E	F	G	H	I
1	Import	INFY EQ	IDBI EQ	ICICIBANK EQ	IBPOW EQ	ICIL EQ	PNB EQ	ELIANCE EQ	
2	09-04-2014 17:45	3505.9	84.8	1443.4	11.25	142.9	926.6	989.75	
3	09-04-2014 17:45	3505	84.8	1443.4	11.3	142.9	926.6	989.4	
4	09-04-2014 17:45	3504.4	84.8	1443.4	11.3	142.9	926.5	989.25	
5	09-04-2014 17:46	3509.55	84.65	1443.75	11.3	142.9	927.6	989.05	
6	09-04-2014 17:50	3509.9	84.7	1444.45	11.35	143.7	927.5	989.15	
7									
8									

**Log:** It allows you to view all the trades that are confirmed from the application and sent to the front-end application; once the order is placed successfully the data will get updated in the Log sheet.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	TIME	Symbol	Market Id	Security Id	Order Mode	Order Type	Order Qty	Order DiscQty	Order Price	Order validity	Product Type	Order Status	
2													
3													
4													
5													
6													
7													
8													
9													
10													

Below is an example of filled logs after the orders are successfully sent to front-end application.

	A	B	C	D	E	F	G	H	I	J	K
1	TIME	Symbol	Market Id	Security Id	Order Mode	Order Type	Order Qty	Order DiscQty	Order Price	Order validity	Product Type
2	09-04-2014 17:45	INFY	3	1594	1	0	1	0	3505.899902	0	0
3	09-04-2014 17:45	IDBI EQ	3	1476	1	0	1	0	84.80000305	0	0
4	09-04-2014 17:45	ICICIBANK EQ	3	4963	1	0	1	0	1443.400024	0	0
5	09-04-2014 17:45	IBPOW EQ	3	17520	1	0	1	0	11.30000019	0	0
6	09-04-2014 17:45	ICIL EQ	3	11987	1	0	1	0	142.8999939	0	0
7	09-04-2014 17:45	PNB EQ	3	10666	1	0	1	0	926.5999756	0	0
8	09-04-2014 17:45	INFY	3	1594	1	0	1	0	3505	0	0
9	09-04-2014 17:45	IDBI EQ	3	1476	1	0	1	0	84.80000305	0	0
10	09-04-2014 17:45	ICICIBANK EQ	3	4963	1	0	1	0	1443.400024	0	0
11	09-04-2014 17:45	IBPOW EQ	3	17520	1	0	1	0	11.30000019	0	0
12	09-04-2014 17:45	ICIL EQ	3	11987	1	0	1	0	142.8999939	0	0
13	09-04-2014 17:45	PNB EQ	3	10666	1	0	1	0	926.5999756	0	0
14	09-04-2014 17:45	INFY	3	1594	1	0	1	0	3505	0	0
15	09-04-2014 17:45	IDBI EQ	3	1476	1	0	1	0	84.69999695	0	0
16	09-04-2014 17:45	ICICIBANK EQ	3	4963	1	0	1	0	1443.400024	0	0
17	09-04-2014 17:45	IBPOW EQ	3	17520	1	0	1	0	11.30000019	0	0
18	09-04-2014 17:45	ICIL EQ	3	11987	1	0	1	0	142.8999939	0	0
19	09-04-2014 17:45	PNB EQ	3	10666	1	0	1	0	926.5999756	0	0
20	09-04-2014 17:46	INFY	3	1594	1	0	1	0	3509.550049	0	0

**OHLC:** When you click on import button placed on top left corner in LTP sheet the symbols in Order sheet automatically gets updated in OHLC sheet as well.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Security Id	Symbol	Interval(Minutes)	Open	High	Low	Close/manual LTP	Volume			Default Interval = 1	
2												
3												
4												
5												
6												
7												
8												
9												
10												

OHLC allows you to define time interval for updating OHLC values for individual scrips. For e.g. if you want to view OHLC values for different scrips at different intervals you may select the scrip and define time interval for getting OHLC values updated.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Security Id	Symbol	Interval(Minutes)	Open	High	Low	Close/manual LT	Volume			Default Interval = 1	
2	1594	INFY EQ	1	3509.89	3510	3509.389893	3509.89	907409				
3	1476	IDBI EQ	2	84.69	84.69000244	84.58999634	84.69	4446309				
4	4963	ICICIBANK EQ	1	1444.44	1444.439941	1442.589966	1444.44	1713887				
5	17520	IBPOW EQ	1	11.35	11.35000038	11.35000038	11.35	1278776				
6	11987	ICIL EQ	4	0	143.6900024	143.6900024	143.69	29623				
7	10666	PNB EQ	2	927.5	927.789978	927.289978	927.5	826461				
8	2885	RELIANCE EQ	1	989.15	989.2000122	989	989.15	2295441				
9												
10												

## 10. Command Line

Command line allows trader to run their own strategies without using the front-end application. Client can directly run the strategy from back-end and deploy own algos.

You need to add following details:

- Server
- Server Port
- Client ID
- Password

Once you receive a successful login message

You will get a main menu as:

- 1) Load New Api
- 2) Start New Api Strategy
- 3) Terminate Strategy
- 4) Terminate All

\*\*\*\*\*

Use numerical to perform various task for e.g. to Start New Api Strategy, you need to input "2". Here is an example of how to run a strategy

->1

Load New Api

Please Enter .so file name

->libfutfut.so

Successfully added strategy libfutfut.so

->2

Start New Api Strategy

Current Loaded Strategy are

1) libfutfut.so

->1

## 11. Special Notes

---

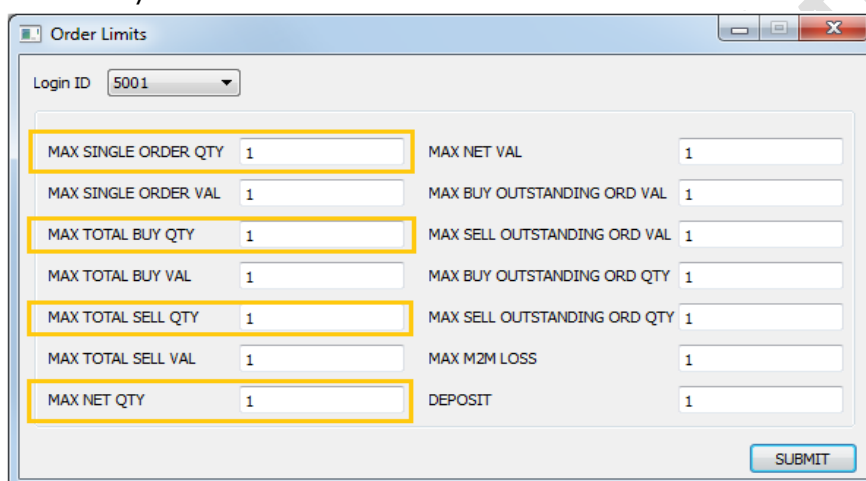
- ✓ Market data shall be subscription based, both TBT and snapshot feed can be subscribed by custom strategy. Latest data shall be provided.
- ✓ SDK type environment with a customer language to write strategy shall NOT be provided
- ✓ C++ API shall only be supported
- ✓ Backend will need to put checks on the data types of strategy parameters

uTrade Solutions Pvt. Ltd.

## 12. Risk Management

### 12.1 Quantity Limits Check

The trading platform validates Order Quantity, and the order generation takes place only if it complies with the Surveillance Measures set for the User with respect to Maximum Single Order Quantity (both Buy/Sell), Maximum Total Order Quantity (both Buy/Sell) and Maximum Net Quantity.



Order Limits

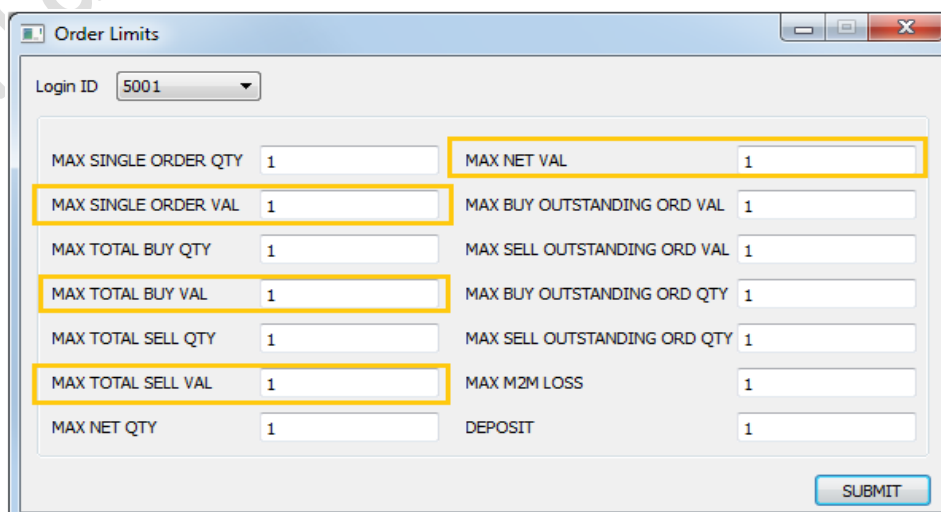
Login ID: 5001

MAX SINGLE ORDER QTY	1	MAX NET VAL	1
MAX SINGLE ORDER VAL	1	MAX BUY OUTSTANDING ORD VAL	1
MAX TOTAL BUY QTY	1	MAX SELL OUTSTANDING ORD VAL	1
MAX TOTAL BUY VAL	1	MAX BUY OUTSTANDING ORD QTY	1
MAX TOTAL SELL QTY	1	MAX SELL OUTSTANDING ORD QTY	1
MAX TOTAL SELL VAL	1	MAX M2M LOSS	1
MAX NET QTY	1	DEPOSIT	1

SUBMIT

### 12.2 Order Value Check

The valid Order Value Generation takes place only if it complies with the Surveillance Measures set for the User with respect to Maximum Single Order Value (both Buy/Sell), Maximum Total Order Value (both Buy/Sell) and Maximum Net Value.



Order Limits

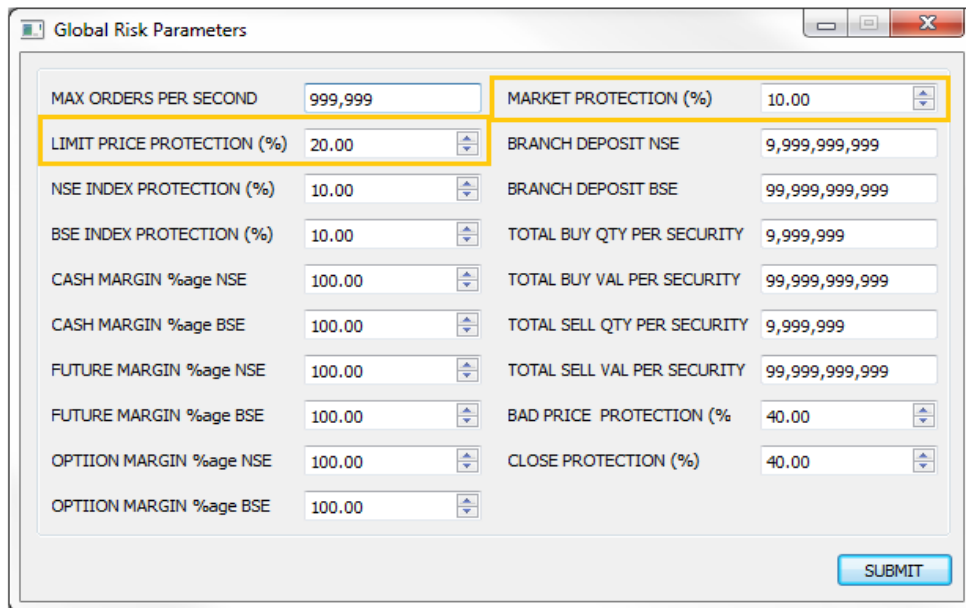
Login ID: 5001

MAX SINGLE ORDER QTY	1	MAX NET VAL	1
MAX SINGLE ORDER VAL	1	MAX BUY OUTSTANDING ORD VAL	1
MAX TOTAL BUY QTY	1	MAX SELL OUTSTANDING ORD VAL	1
MAX TOTAL BUY VAL	1	MAX BUY OUTSTANDING ORD QTY	1
MAX TOTAL SELL QTY	1	MAX SELL OUTSTANDING ORD QTY	1
MAX TOTAL SELL VAL	1	MAX M2M LOSS	1
MAX NET QTY	1	DEPOSIT	1

SUBMIT

## 12.3 Price Range check

The order price generation mechanism strictly follows the specified Range provided to a trader to limit the risk within a pre-set percentage of the Last Trade Price (LTP). The percentage will be greater than zero and less than or equal to the applicable circuit limits including dummy circuit limits in respect of all algorithmic/single orders.

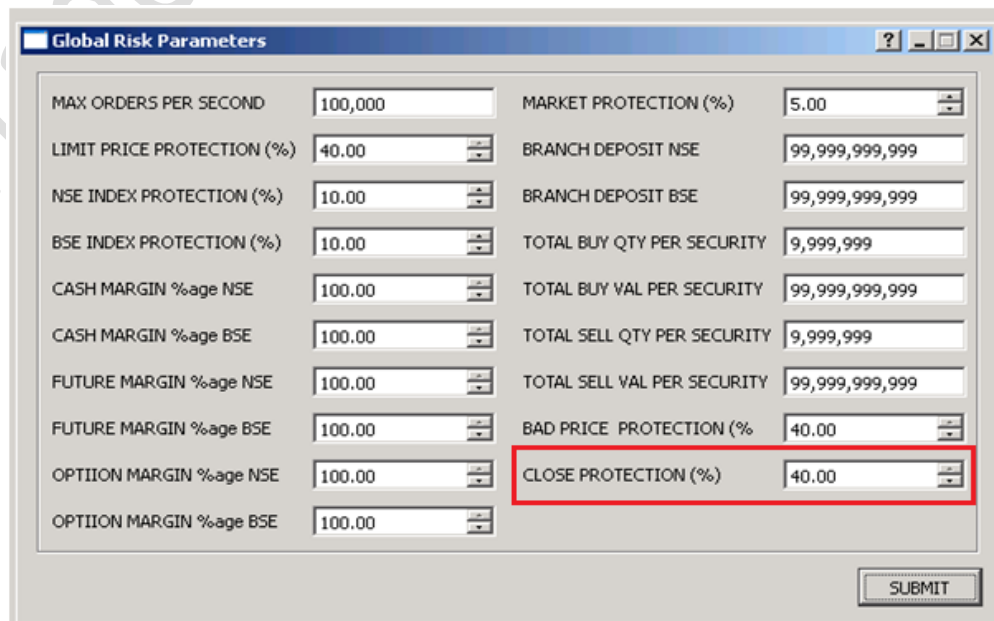


Global Risk Parameters	
MAX ORDERS PER SECOND	999,999
MARKET PROTECTION (%)	10.00
LIMIT PRICE PROTECTION (%)	20.00
NSE INDEX PROTECTION (%)	10.00
BSE INDEX PROTECTION (%)	10.00
CASH MARGIN %age NSE	100.00
CASH MARGIN %age BSE	100.00
FUTURE MARGIN %age NSE	100.00
FUTURE MARGIN %age BSE	100.00
OPTION MARGIN %age NSE	100.00
OPTION MARGIN %age BSE	100.00
BRANCH DEPOSIT NSE	9,999,999,999
BRANCH DEPOSIT BSE	99,999,999,999
TOTAL BUY QTY PER SECURITY	9,999,999
TOTAL BUY VAL PER SECURITY	99,999,999,999
TOTAL SELL QTY PER SECURITY	9,999,999
TOTAL SELL VAL PER SECURITY	99,999,999,999
BAD PRICE PROTECTION (%)	40.00
CLOSE PROTECTION (%)	40.00

SUBMIT

## 12.4 Closing Price Check

The order price generation mechanism strictly follows the specified Range provided to a trader to limit the risk within a pre-set percentage of the Closing Price.



Global Risk Parameters	
MAX ORDERS PER SECOND	100,000
MARKET PROTECTION (%)	5.00
LIMIT PRICE PROTECTION (%)	40.00
NSE INDEX PROTECTION (%)	10.00
BSE INDEX PROTECTION (%)	10.00
CASH MARGIN %age NSE	100.00
CASH MARGIN %age BSE	100.00
FUTURE MARGIN %age NSE	100.00
FUTURE MARGIN %age BSE	100.00
OPTION MARGIN %age NSE	100.00
OPTION MARGIN %age BSE	100.00
BRANCH DEPOSIT NSE	99,999,999,999
BRANCH DEPOSIT BSE	99,999,999,999
TOTAL BUY QTY PER SECURITY	9,999,999
TOTAL BUY VAL PER SECURITY	99,999,999,999
TOTAL SELL QTY PER SECURITY	9,999,999
TOTAL SELL VAL PER SECURITY	99,999,999,999
BAD PRICE PROTECTION (%)	40.00
CLOSE PROTECTION (%)	40.00

SUBMIT



## 12.5 Trade Price Protection Check (Bad trade Protection)

By defining a field value in Bad Price Protection % Change, order price generation can be restricted within a set percentage of last two LTP (Last trade price). Algo trade generation beyond set price bracket will result into rejection/freeze of the respective Algo rule until consecutive LTP received is within the set % range.

Global Risk Parameters			
MAX ORDERS PER SECOND	100,000	MARKET PROTECTION (%)	5.00
LIMIT PRICE PROTECTION (%)	40.00	BRANCH DEPOSIT NSE	99,999,999,999
NSE INDEX PROTECTION (%)	10.00	BRANCH DEPOSIT BSE	99,999,999,999
BSE INDEX PROTECTION (%)	10.00	TOTAL BUY QTY PER SECURITY	9,999,999
CASH MARGIN %age NSE	100.00	TOTAL BUY VAL PER SECURITY	99,999,999,999
CASH MARGIN %age BSE	100.00	TOTAL SELL QTY PER SECURITY	9,999,999
FUTURE MARGIN %age NSE	100.00	TOTAL SELL VAL PER SECURITY	99,999,999,999
FUTURE MARGIN %age BSE	100.00	<b>BAD PRICE PROTECTION (%)</b>	<b>40.00</b>
OPTION MARGIN %age NSE	100.00	CLOSE PROTECTION (%)	40.00
OPTION MARGIN %age BSE	100.00		

SUBMIT

## 12.6 Position Level Check

Position limit check at Client/User level can be established by defining the maximum permissible value in Net Quantity Field.

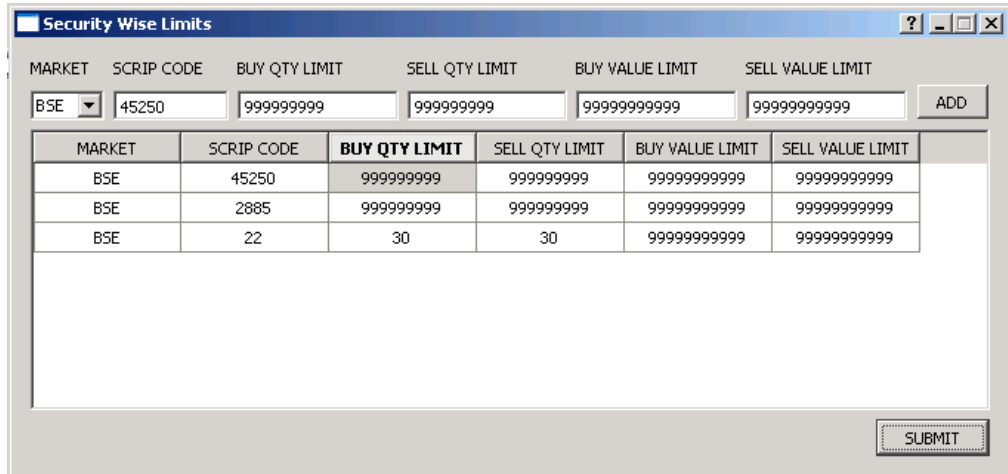
Order Limits			
Login ID	5001		
MAX SINGLE ORDER QTY	1	MAX NET VAL	1
MAX SINGLE ORDER VAL	1	MAX BUY OUTSTANDING ORD VAL	1
MAX TOTAL BUY QTY	1	MAX SELL OUTSTANDING ORD VAL	1
MAX TOTAL BUY VAL	1	MAX BUY OUTSTANDING ORD QTY	1
MAX TOTAL SELL QTY	1	MAX SELL OUTSTANDING ORD QTY	1
MAX TOTAL SELL VAL	1	MAX M2M LOSS	1
<b>MAX NET QTY</b>	<b>1</b>	DEPOSIT	1

SUBMIT



## 12.7 Security wise limit

The checks validates the buy/sell quantity & value security wise & prevent the user from entering of any fresh orders in case specified limit is reached.



The screenshot shows a window titled "Security Wise Limits" with a table of limits. The table has columns for MARKET, SCRIP CODE, BUY QTY LIMIT, SELL QTY LIMIT, BUY VALUE LIMIT, and SELL VALUE LIMIT. The first three rows show limits for BSE market with SCRIP CODE 45250, 2885, and 22. The first two rows have limits of 999999999, while the third row has limits of 30. The fourth row is empty. There are input fields at the top for each column and an "ADD" button. A "SUBMIT" button is at the bottom right.

MARKET	SCRIP CODE	BUY QTY LIMIT	SELL QTY LIMIT	BUY VALUE LIMIT	SELL VALUE LIMIT
BSE	45250	999999999	999999999	999999999999	999999999999
BSE	2885	999999999	999999999	999999999999	999999999999
BSE	22	30	30	999999999999	999999999999

## 13. Compliance Requirements

---

### 13.1 Order Identification

The orders generated through Algo Platform are identified with unique order number as per the exchange criteria.

### 13.2 Non- Tamper able Audit Trial

Access to system is restricted to authorized person permitted by the system administrator. System has provision to create daily backup for trade log, Message log, and User maintenance back up, Order history back up, etc.

### 13.3 Orders per second

Maximum number of Order released per second by the Algo platform is in accordance with the exchange criteria.

### 13.4 Automated Execution Throttle

The Algo platform has In-built functions to ensure that next set of orders are generated only after fulfilment of first phase of Trade cycle.

## 13.5 System security

Access to system is restricted to authorized person permitted by the system administrator with valid User Id and Password.

Some more characteristics are:

- ✓ *Automatic disablement of the user on entering erroneous password on three consecutive occasions.*
- ✓ *Automatic expiry of password on expiry of 14 calendar days.*
- ✓ *The Passwords are alphanumeric, instead of just being alphabets or just numerical.*
- ✓ *System doesn't allow the changed password to be the same as the last password*
- ✓ *Login id of the user and password cannot be the same.*
- ✓ *Minimum password length is eight characters and not more than twelve characters.*
- ✓ *The Passwords are stored in encrypted form to safe guard against any misuse*

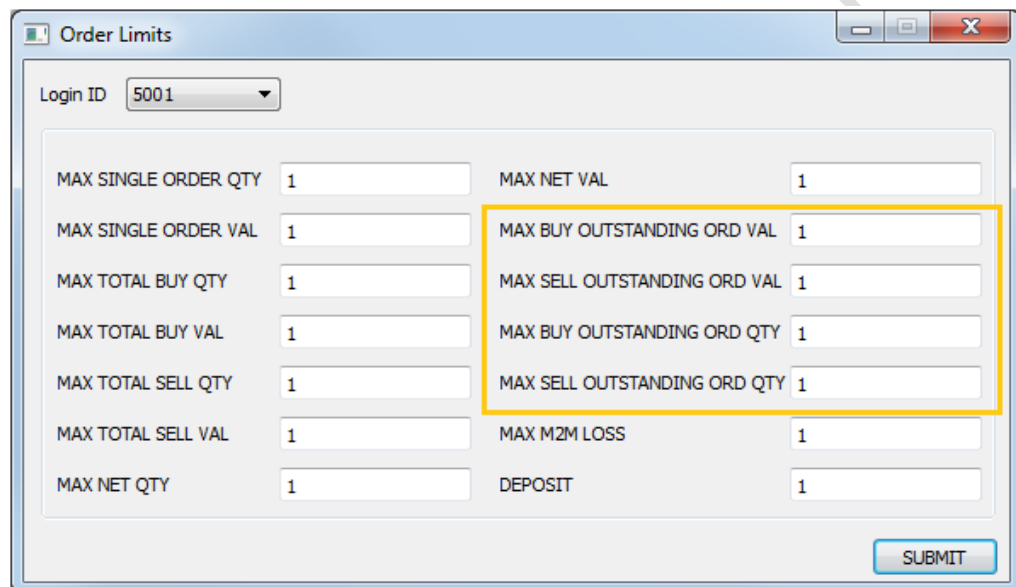
## 13.6 Control Values

Audit trail created in our system captures the record of control parameters, orders, trades and data points emanating from trades executed through algorithm trading. Field values are made mandatory for all risk control parameters, and no risk parameters have provision to accept unlimited value or Zero value.

Before sending any order to exchange trading system, the orders are mandatorily filtered through the Surveillance system. Those orders which are within the set parameters will only flow to the exchange.

## 13.7 Dysfunctional Algos

- ✓ Surveillance measures to check total cumulative Buy/Sell Quantity, Total Net Quantity are built into the system to restrict/identify position building on one side or overall basis. Further orders in excess of set standards of Surveillance will freeze/rejected. System has functions to restrict the entry on number of rules.



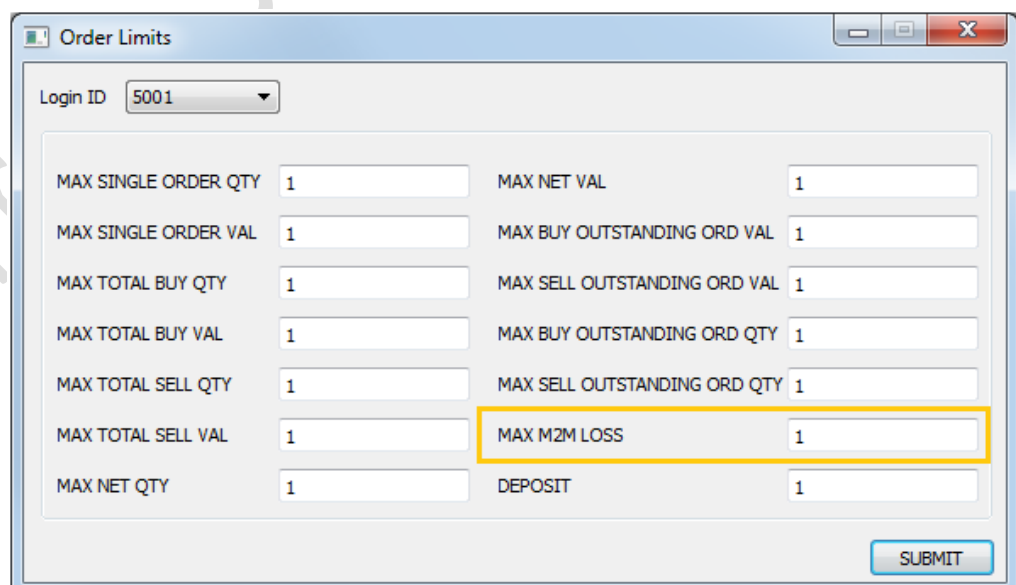
Order Limits

Login ID: 5001

MAX SINGLE ORDER QTY	1	MAX NET VAL	1
MAX SINGLE ORDER VAL	1	MAX BUY OUTSTANDING ORD VAL	1
MAX TOTAL BUY QTY	1	MAX SELL OUTSTANDING ORD VAL	1
MAX TOTAL BUY VAL	1	MAX BUY OUTSTANDING ORD QTY	1
MAX TOTAL SELL QTY	1	MAX SELL OUTSTANDING ORD QTY	1
MAX TOTAL SELL VAL	1	MAX M2M LOSS	1
MAX NET QTY	1	DEPOSIT	1

SUBMIT

- ✓ The algorithm shall pause if the Maximum MTM loss crosses the specified limit



Order Limits

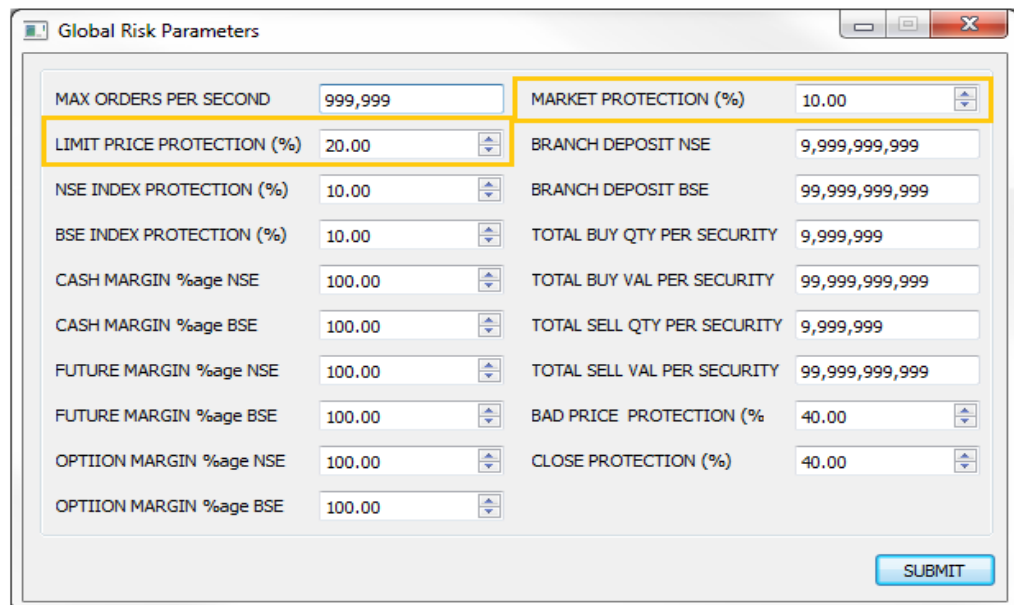
Login ID: 5001

MAX SINGLE ORDER QTY	1	MAX NET VAL	1
MAX SINGLE ORDER VAL	1	MAX BUY OUTSTANDING ORD VAL	1
MAX TOTAL BUY QTY	1	MAX SELL OUTSTANDING ORD VAL	1
MAX TOTAL BUY VAL	1	MAX BUY OUTSTANDING ORD QTY	1
MAX TOTAL SELL QTY	1	MAX SELL OUTSTANDING ORD QTY	1
MAX TOTAL SELL VAL	1	MAX M2M LOSS	1
MAX NET QTY	1	DEPOSIT	1

SUBMIT

## 13.8 Market Price Protection

By defining a field value in Price % Change, order price generation can be restricted within a set percentage range of LTP. Algo price generation beyond set price bracket will result into rejection/freeze of the respective Algo rule.



The screenshot shows the 'Global Risk Parameters' window with the following settings:

Parameter	Value
MAX ORDERS PER SECOND	999,999
MARKET PROTECTION (%)	10.00
LIMIT PRICE PROTECTION (%)	20.00
BRANCH DEPOSIT NSE	9,999,999,999
NSE INDEX PROTECTION (%)	10.00
BRANCH DEPOSIT BSE	99,999,999,999
BSE INDEX PROTECTION (%)	10.00
TOTAL BUY QTY PER SECURITY	9,999,999
CASH MARGIN %age NSE	100.00
TOTAL BUY VAL PER SECURITY	99,999,999,999
CASH MARGIN %age BSE	100.00
TOTAL SELL QTY PER SECURITY	9,999,999
FUTURE MARGIN %age NSE	100.00
TOTAL SELL VAL PER SECURITY	99,999,999,999
FUTURE MARGIN %age BSE	100.00
BAD PRICE PROTECTION (%)	40.00
OPTION MARGIN %age NSE	100.00
CLOSE PROTECTION (%)	40.00
OPTION MARGIN %age BSE	100.00

A 'SUBMIT' button is located at the bottom right of the window.

# 14. System Requirements

## HARDWARE AND SOFTWARE REQUIREMENTS

Hardware/Software	Minimum	Recommended
BACK END		
Processor	4 Core CPU	12 Core CPU
RAM	8 GB	32 GB
Operating System	Linux OS 64-bit (Redhat/CentOS 6.0 / 6.4)	
FRONT END		
Processor	2 Core CPU	4 Core CPU
RAM	4 GB	8 GB
Operating System	Windows Server	

## Contact Us

---



We would love to hear from you to understand your business needs and discuss how we could assist you in achieving your goals. Please contact us at [info@utradesolutions.com](mailto:info@utradesolutions.com) for further details.

uTrade Solutions Pvt. Ltd.



## uTrade Solutions Private Ltd

2nd floor, landmark plaza (f3). Quarkcity SEZ, A40A,  
Industrial Area, Phase 8 extension,  
Mohali, Chandigarh 160071 (India)  
contact: [info@utradesolutions.com](mailto:info@utradesolutions.com)