uTrade
Enabling Smarter Trading

www.utradesolutions.com

uTrade Solutions

API Version 2.0 Client Manual

(Latency Sensitive)

TRADING
ANALYTICS
CUSTOM SOLUTIONS
ALGOS

Table of Contents

Table of Contents 2

Disclaimer 3

1. **Company Overview** 4

2. **Preface** 5

3. **Scope of Document** 5

4. **Overview** 5

5. **Steps for writing and running custom strategy** 6

6. **Designing Front-end for Custom Strategy** 6

7. **Steps for writing a strategy (back-end implementation)** 17

8 **Things to do before calling Start Algo** 20

9 **Special Notes** 21

10 **Risk Management** 22

11 **Compliance Requirements** 25

12 **System Requirements** 28

13 **Contact Us** 29

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.

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, does not cover the in-built strategies which come with the uTrade platform.

4. Overview

- ✓ uTrade's API 2.0 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 supports all segments of NSE, BSE, SGX, CFH

5. Steps for writing and running custom strategy

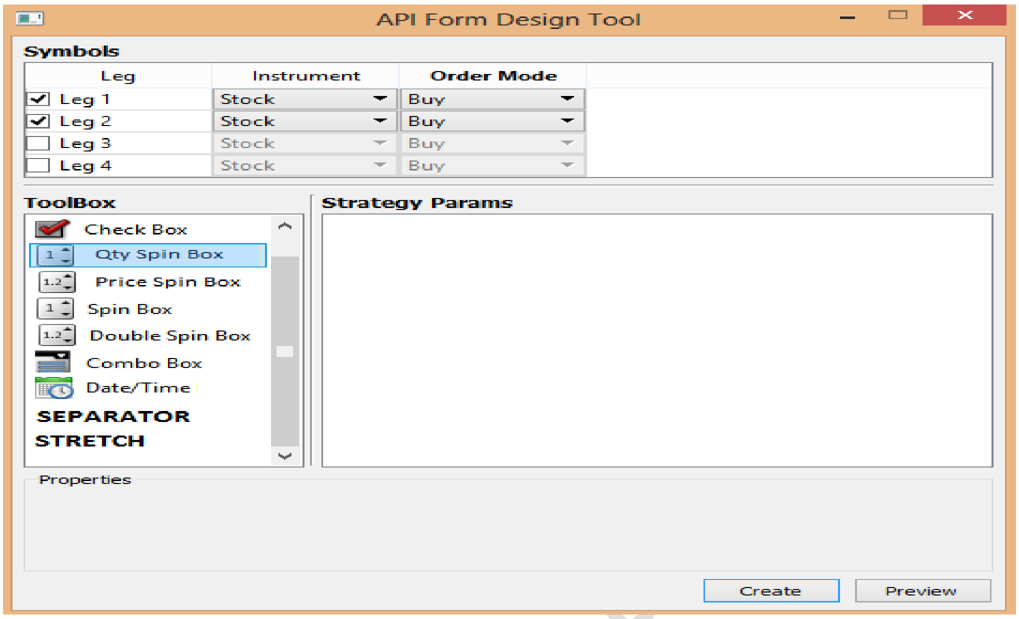
- ✓ You need to generate front-end using an in-built tool
- ✓ You need to code and create custom strategy .so file (as shared object or dynamic library)
- ✓ At back-end, you need to copy the .so to custom strategy folder.
- ✓ Now, to run the strategy from front-end, you need to specify the name of .so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

6. Designing Front-end for Custom Strategy

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

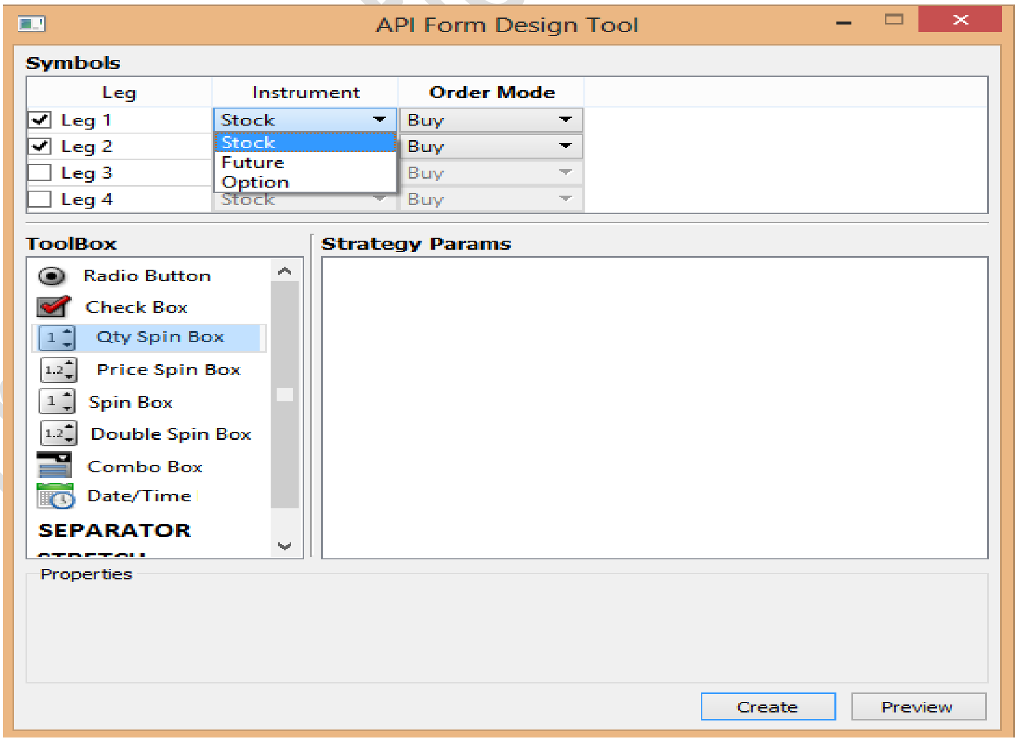
Depending on the type of parameter, front-end form gets generated where you can fill values based on logic, and integrate with the custom strategy back-end implementation.

6.1 Tool Box


API Form Design Tool
Symbols
ToolBox
Check Box
Qty Spin Box
Price Spin Box
Spin Box
Double Spin Box
Combo Box
Date/Time
SEPARATOR
STRETCH
Strategy Params
Properties
Create
Preview

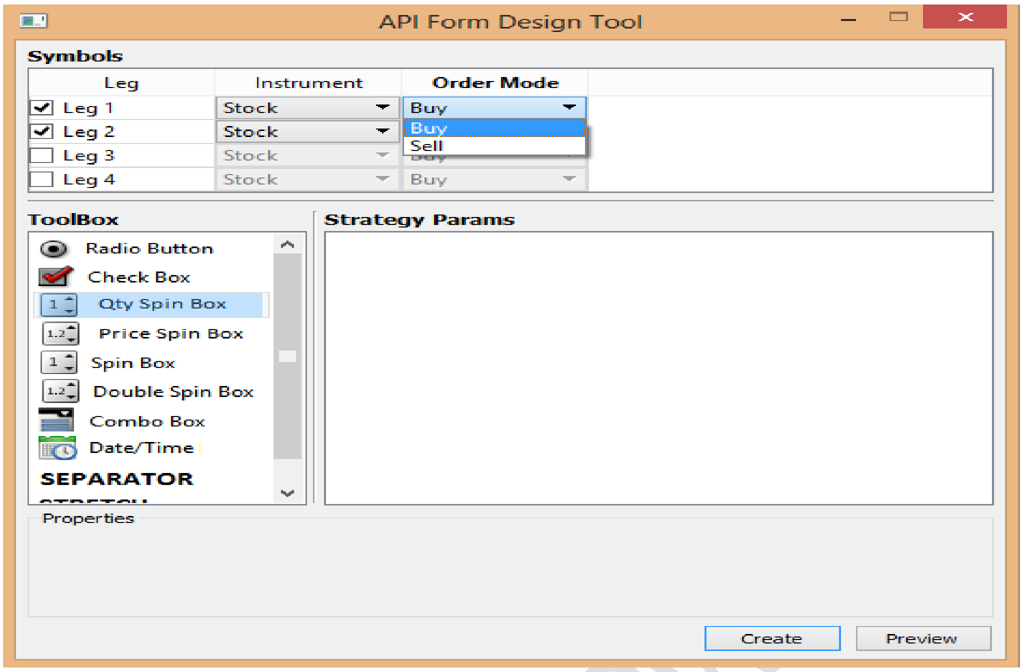
Leg	Instrument	Order Mode
☑ Leg 1	Stock	Buy
☑ Leg 2	Stock	Buy
☐ Leg 3	Stock	Buy
☐ Leg 4	Stock	Buy

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.


API Form Design Tool
Symbols
Stock
Future
Option
ToolBox
Radio Button
Check Box
Qty Spin Box
Price Spin Box
Spin Box
Double Spin Box
Combo Box
Date/Time
SEPARATOR
Strategy Params
Properties
Create
Preview

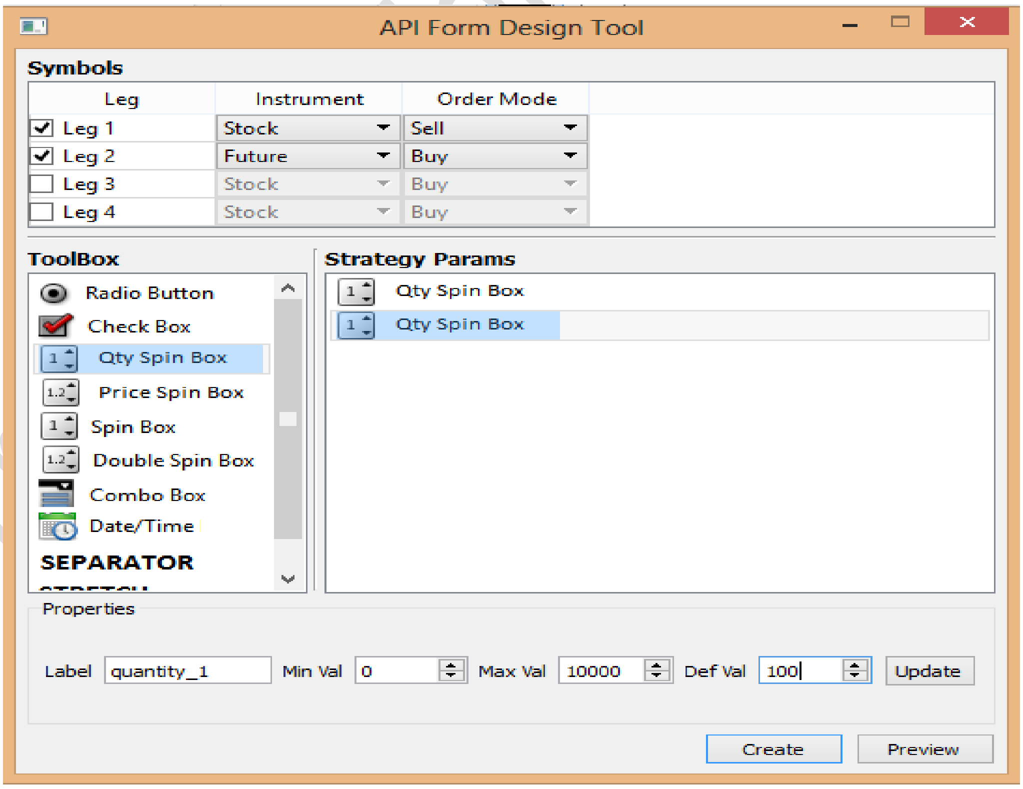
Leg	Instrument	Order Mode
☑ Leg 1	Stock	Buy
☑ Leg 2		Buy
☐ Leg 3		Buy
☐ Leg 4	Stock	Buy

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


API Form Design Tool
Symbols
ToolBox
Radio Button
Check Box
Qty Spin Box
Price Spin Box
Spin Box
Double Spin Box
Combo Box
Date/Time
SEPARATOR
Strategy Params
Properties
Create
Preview

Leg	Instrument	Order Mode
☑ Leg 1	Stock	Buy
☑ Leg 2	Stock	Buy Sell
☐ Leg 3	Stock	Buy
☐ Leg 4	Stock	Buy

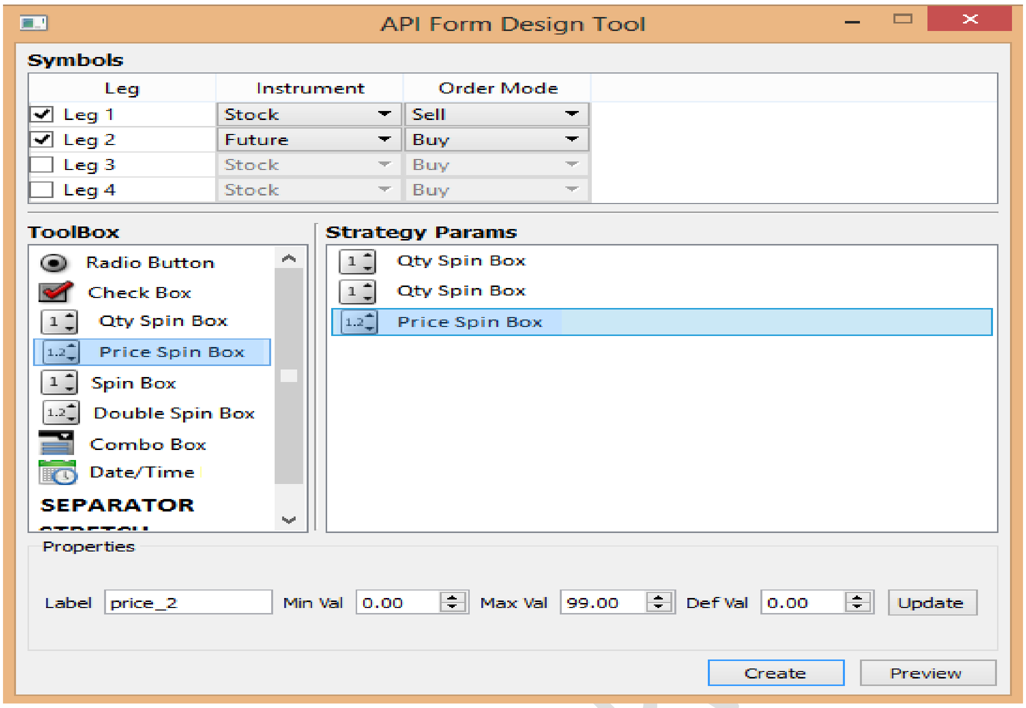
Select, buy or sell options for different legs. For e.g. if you want to sell cash and buy futures, select sell for leg one and buy for second leg.


API Form Design Tool
Symbols
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
Update
Create
Preview

Leg	Instrument	Order Mode
☑ Leg 1	Stock	Sell
☑ Leg 2	Future	Buy
☐ Leg 3	Stock	Buy
☐ Leg 4	Stock	Buy

Label quantity_1 Min Val 0 Max Val 10000 Def Val 100

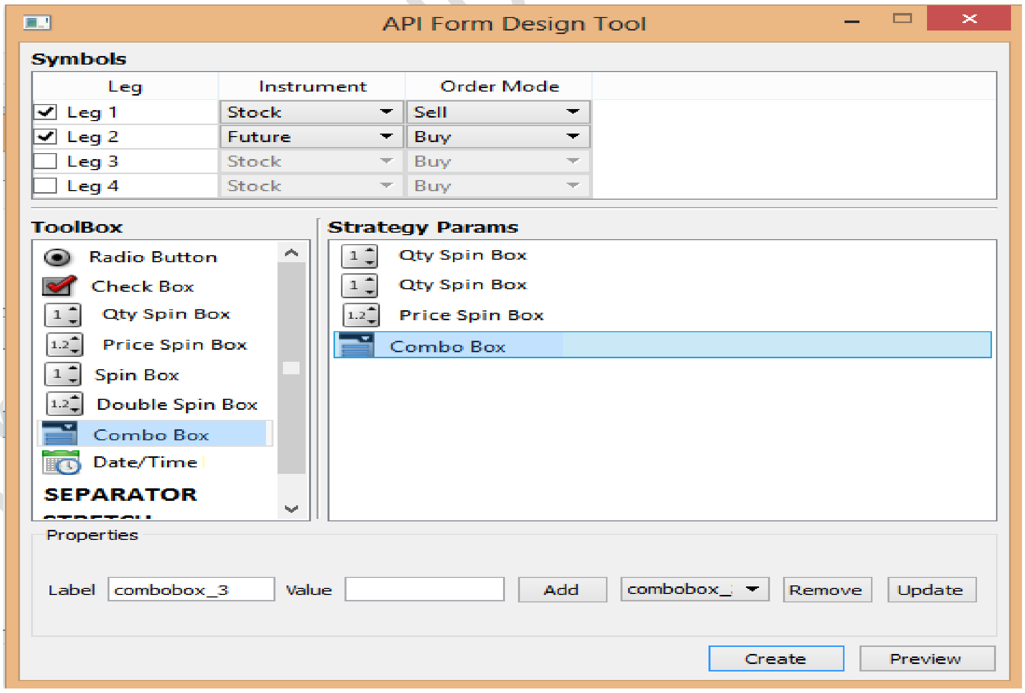
Select Spin box for entering quantity or whole values. Define label like Order Quantity, define value range min and max, enter default value you want to trade on and update.


API Form Design Tool
Symbols
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
Update
Create
Preview

Leg	Instrument	Order Mode
☑ Leg 1	Stock	Sell
☑ Leg 2	Future	Buy
☐ Leg 3	Stock	Buy
☐ Leg 4	Stock	Buy

Label price_2 Min Val 0.00 Max Val 99.00 Def Val 0.00

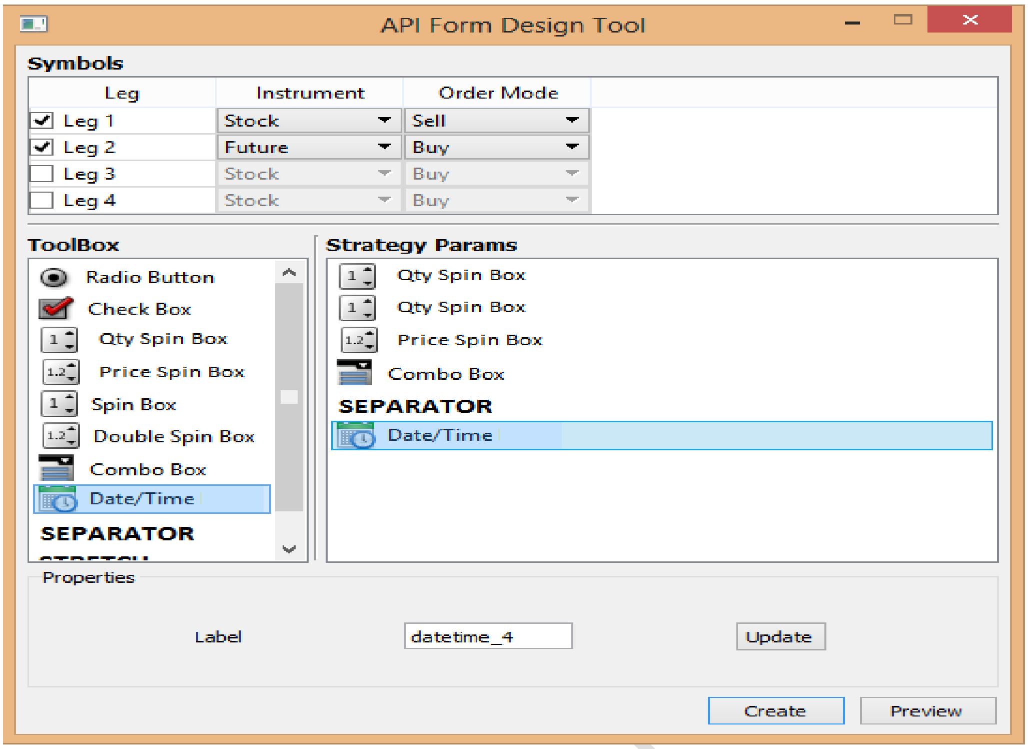
Select double spin box for entering price or rational numbers. Define label like Min Profit, define price range min and max, set default price you want to trade on and update.


API Form Design Tool
Symbols
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
Combo Box
Properties
Add
Remove
Update
Create
Preview

Leg	Instrument	Order Mode
☑ Leg 1	Stock	Sell
☑ Leg 2	Future	Buy
☐ Leg 3	Stock	Buy
☐ Leg 4	Stock	Buy

Label combobox_3 Value combobox_

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.

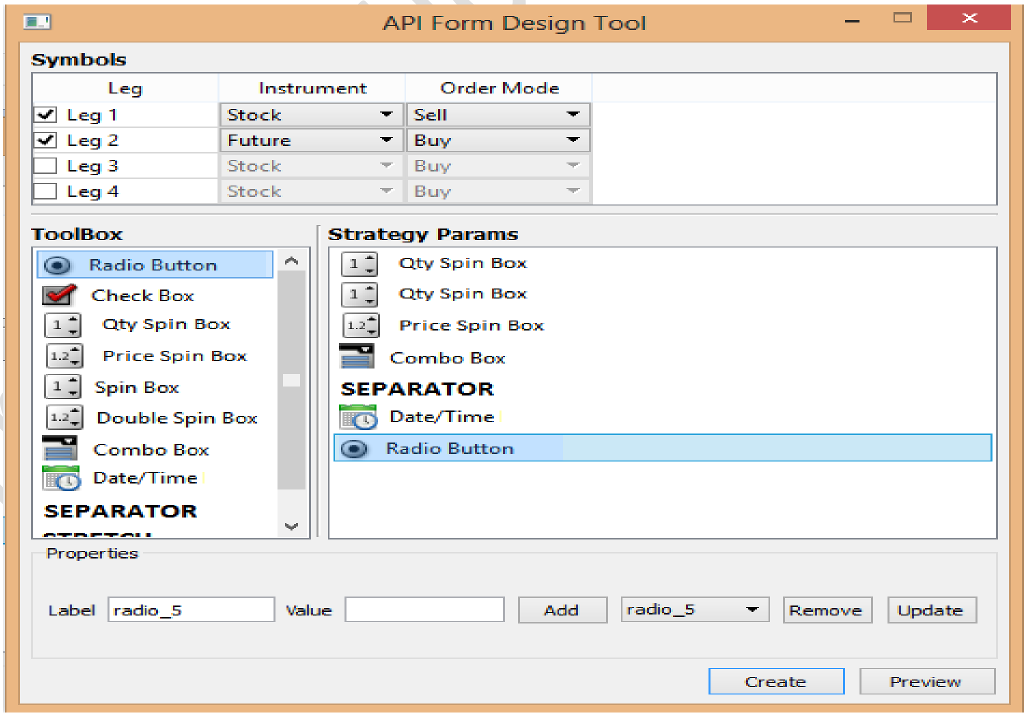


Leg	Instrument	Order Mode
☑ Leg 1	Stock	Sell
☑ Leg 2	Future	Buy
☐ Leg 3	Stock	Buy
☐ Leg 4	Stock	Buy

Label: datetime_4

API Form Design Tool
Symbols
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
Combo Box
SEPARATOR
Date/Time
Properties
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.

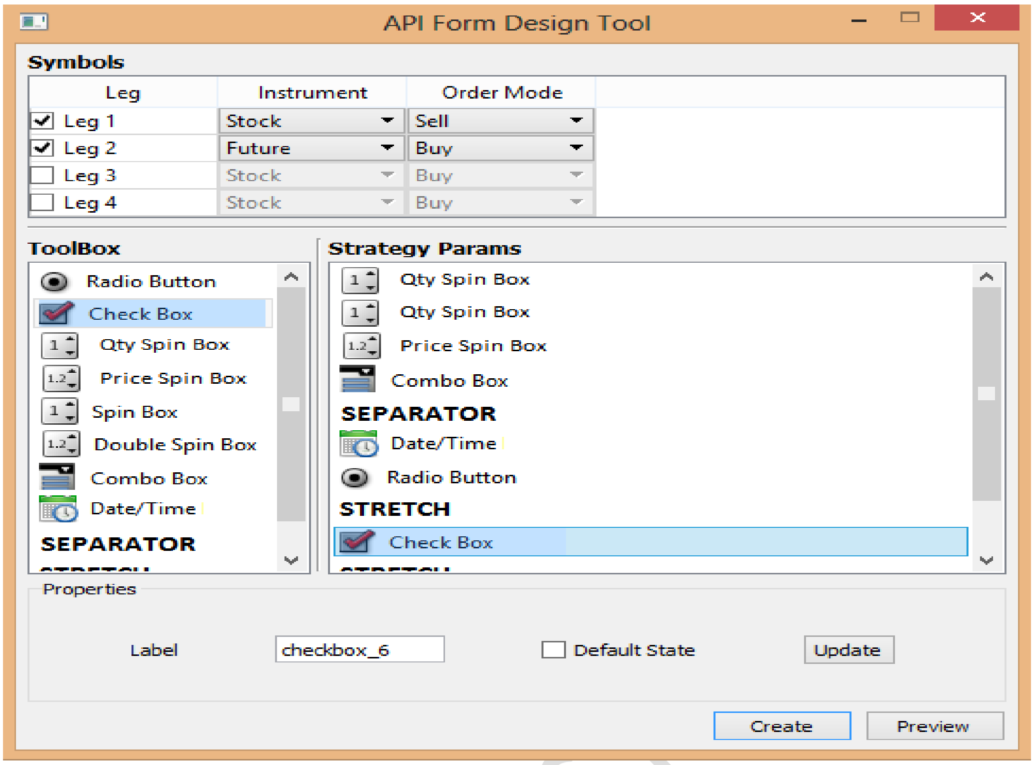


Leg	Instrument	Order Mode
☑ Leg 1	Stock	Sell
☑ Leg 2	Future	Buy
☐ Leg 3	Stock	Buy
☐ Leg 4	Stock	Buy

Label: radio_5 Value: radio_5

API Form Design Tool
Symbols
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
Combo Box
SEPARATOR
Date/Time
Radio Button
Properties
Add
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.

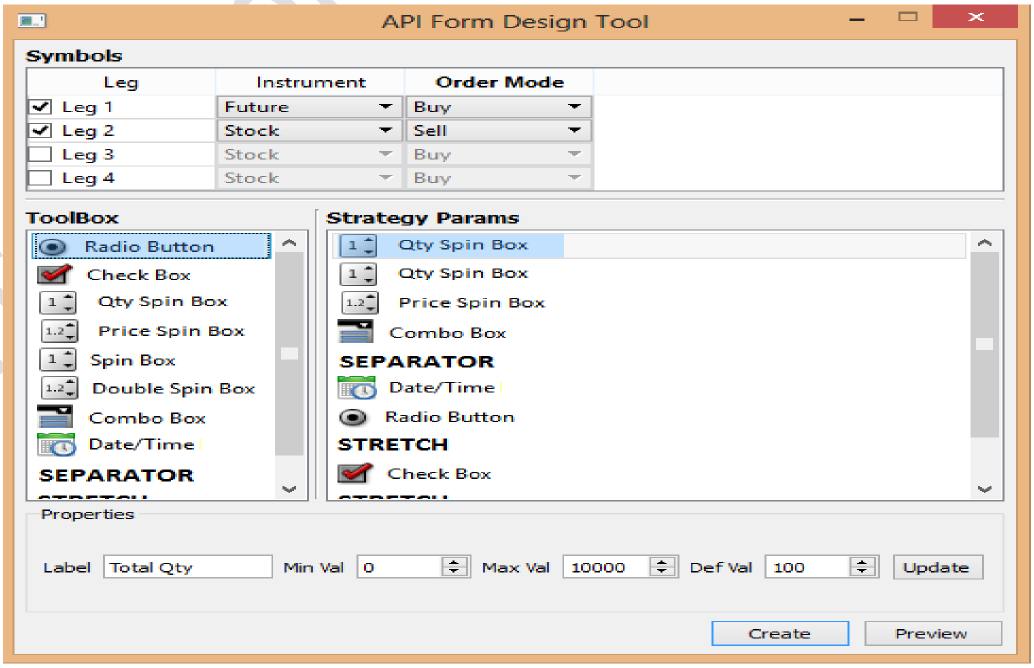

API Form Design Tool
Symbols
ToolBox
Radio Button
Check Box
Qty Spin Box
Price Spin Box
Spin Box
Double Spin Box
Combo Box
Date/Time
SEPARATOR
Strategy Params
Properties
Create
Preview

Leg	Instrument	Order Mode
☑ Leg 1	Stock	Sell
☑ Leg 2	Future	Buy
☐ Leg 3	Stock	Buy
☐ Leg 4	Stock	Buy

Strategy Params: Qty Spin Box, Qty Spin Box, Price Spin Box, Combo Box, SEPARATOR, Date/Time, Radio Button, STRETCH, Check Box

Label checkbox_6 ☐ Default State Update

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.

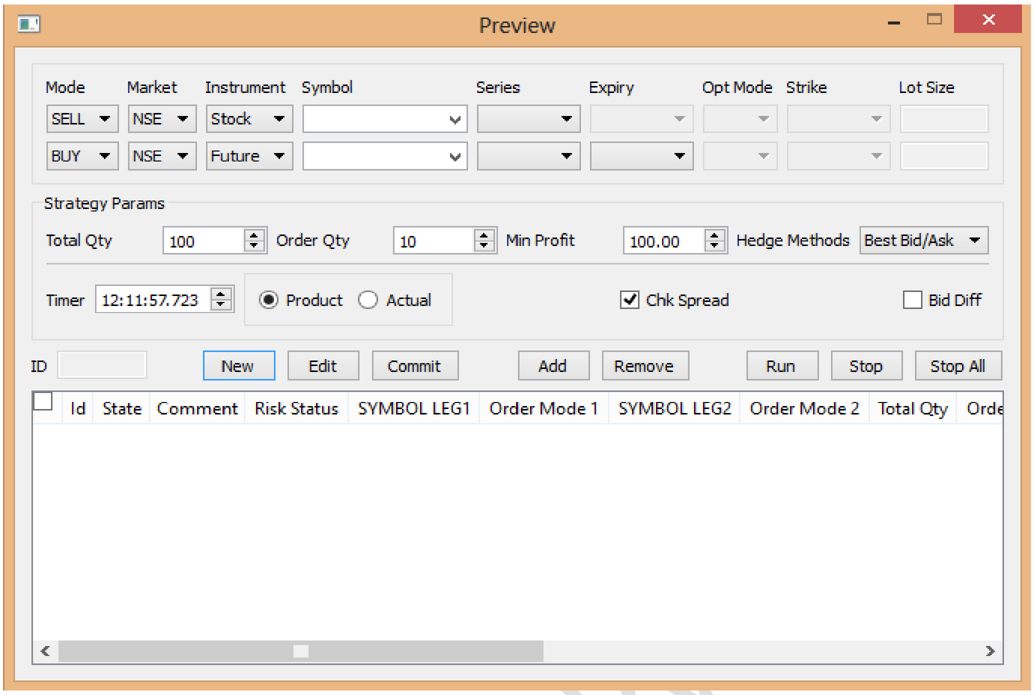

API Form Design Tool
Symbols
ToolBox
Radio Button
Check Box
Qty Spin Box
Price Spin Box
Spin Box
Double Spin Box
Combo Box
Date/Time
SEPARATOR
Strategy Params
Properties
Create
Preview

Leg	Instrument	Order Mode
☑ Leg 1	Future	Buy
☑ Leg 2	Stock	Sell
☐ Leg 3	Stock	Buy
☐ Leg 4	Stock	Buy

Strategy Params: Qty Spin Box, Qty Spin Box, Price Spin Box, Combo Box, SEPARATOR, Date/Time, Radio Button, STRETCH, Check Box

Label Total Qty Min Val 0 Max Val 10000 Def Val 100 Update

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


Preview
New
Edit
Commit
Add
Remove
Run
Stop
Stop All

Mode	Market	Instrument	Symbol	Series	Expiry	Opt Mode	Strike	Lot Size
SELL	NSE	Stock						
BUY	NSE	Future						

Strategy Params

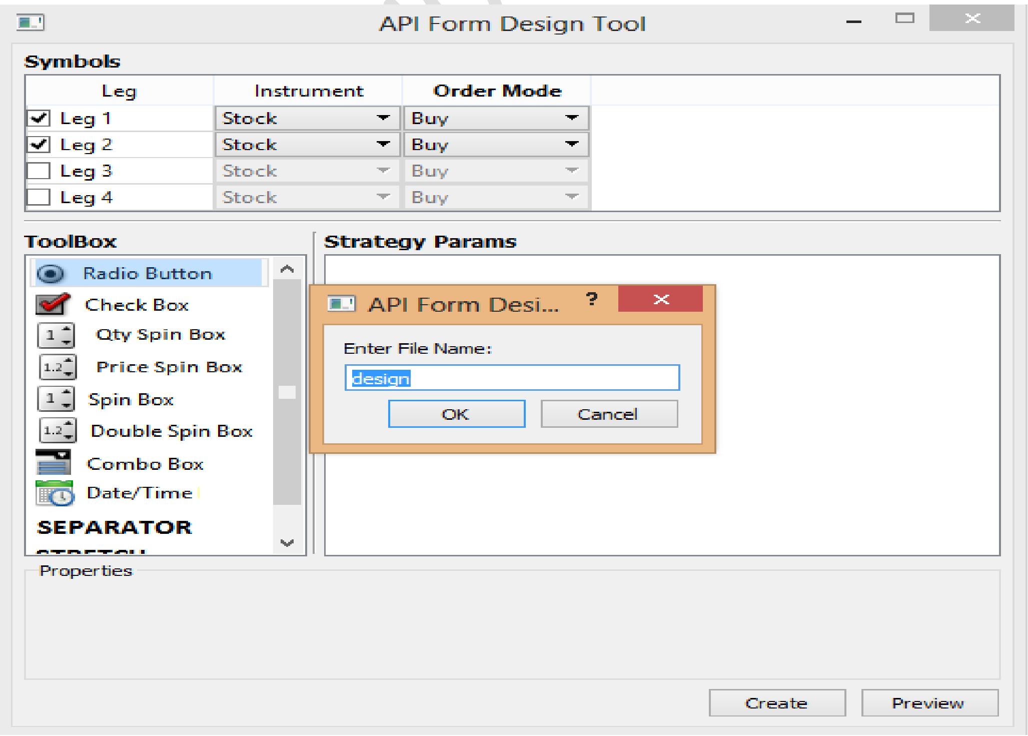
Total Qty 100 Order Qty 10 Min Profit 100.00 Hedge Methods Best Bid/Ask

Timer 12:11:57.723 ● Product ○ Actual ☑ Chk Spread ☐ Bid Diff

ID

☐	Id	State	Comment	Risk Status	SYMBOL LEG1	Order Mode 1	SYMBOL LEG2	Order Mode 2	Total Qty	Orde

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.


API Form Design Tool
ToolBox
Radio Button
Check Box
Qty Spin Box
Price Spin Box
Spin Box
Double Spin Box
Combo Box
Date/Time
SEPARATOR
Strategy Params
Properties
OK
Cancel
Create
Preview

Symbols

Leg	Instrument	Order Mode
☑ Leg 1	Stock	Buy
☑ Leg 2	Stock	Buy
☐ Leg 3	Stock	Buy
☐ Leg 4	Stock	Buy

API Form Desi...

Enter File Name: design

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:

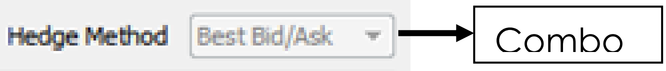
```
[SYMBOL]
SYMBOL LEG1=UINT64:F
Order Mode 1=UCHAR:1
SYMBOL LEG2=UINT64:F
Order Mode 2=UCHAR:0
[STRATEGY_PARAMS]
Total Qty=UINT64:SPINBOX:0:99999:1:Q
Order Qty=UINT64:SPINBOX:0:99999:1:Q
Min Profit=FLOAT:DSPINBOX:0.00:999.00:0.05:P
Hedge Method=COMBO:Market Order:Best Bid/Ask:Best Ask/Bid:Best Bid/Ask
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
```



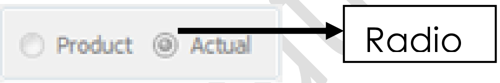
Hedge Method Best Bid/Ask

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
```

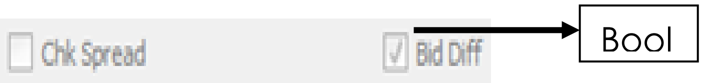
Last value (radio 2) is default checked radio button.



○ Product ● Actual

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.

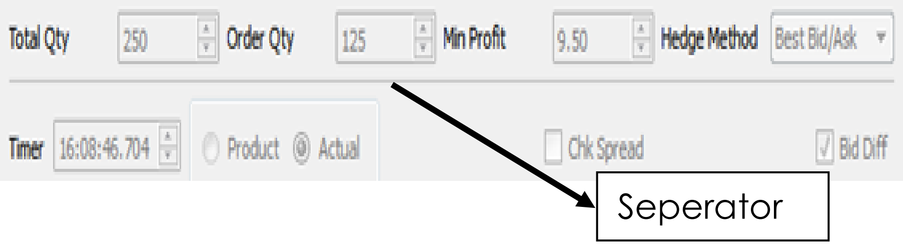
6.2.3 CHECK BOX



☐ Chk Spread ☑ Bid Diff

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

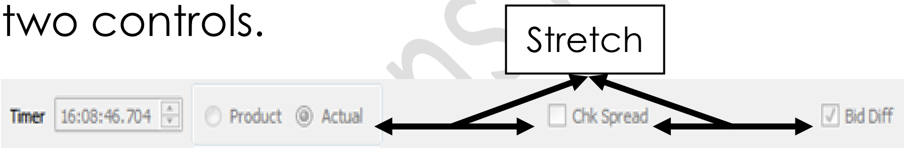
6.2.4 SEPARATOR will create a horizontal line



Total Qty 250
Order Qty 125
Min Profit 9.50
Hedge Method Best Bid/Ask
Timer 16:08:46.704
○ Product ● Actual
☐ Chk Spread
☑ Bid Diff

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.



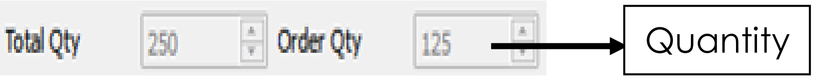
Timer 16:08:46.704
○ Product ● Actual
☐ Chk Spread
☑ Bid Diff

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.



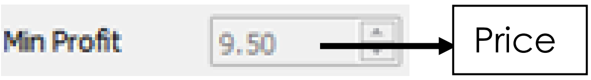
Total Qty 250
Order Qty 125

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.



Min Profit 9.50

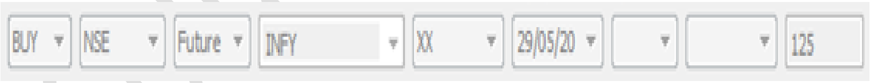
Price

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 you 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::processCurrentState)

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::startAlgo()

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::startAlgo()

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::addInstrument or
  API2::SGContext::getAddInstrument
- 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::startAlgo** 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 an whenever feed change is detected event callback

  onDerivedMarketDataEvent 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::processCurrentState** is generated infinitely. The algo is required to update Market Data, by calling API2::SGContext::updateQuote() to update market data.

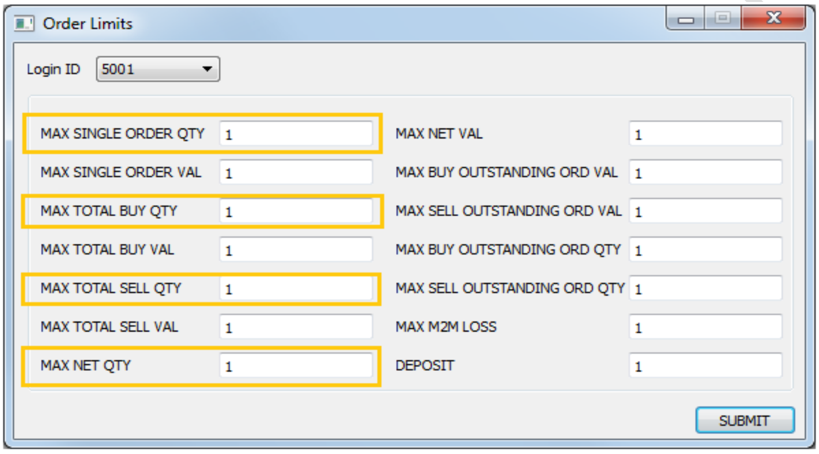
9 Special Notes

- ✓ 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

10 Risk Management

10.1 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.

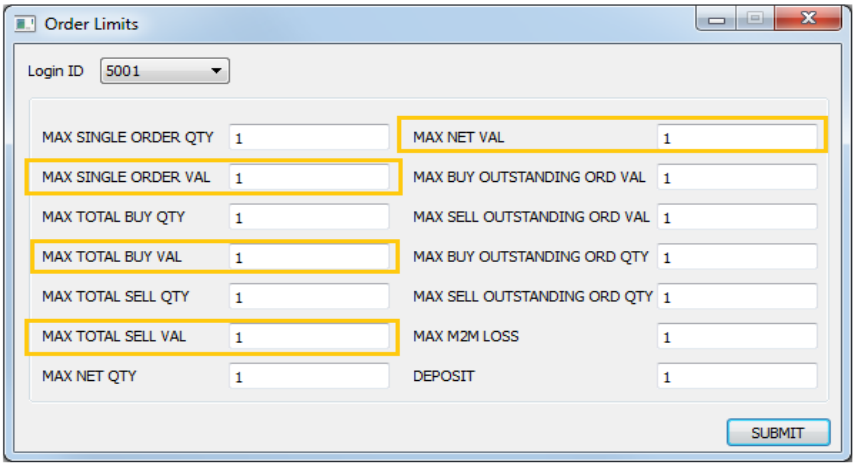


Login ID 5001

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

10.2 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.

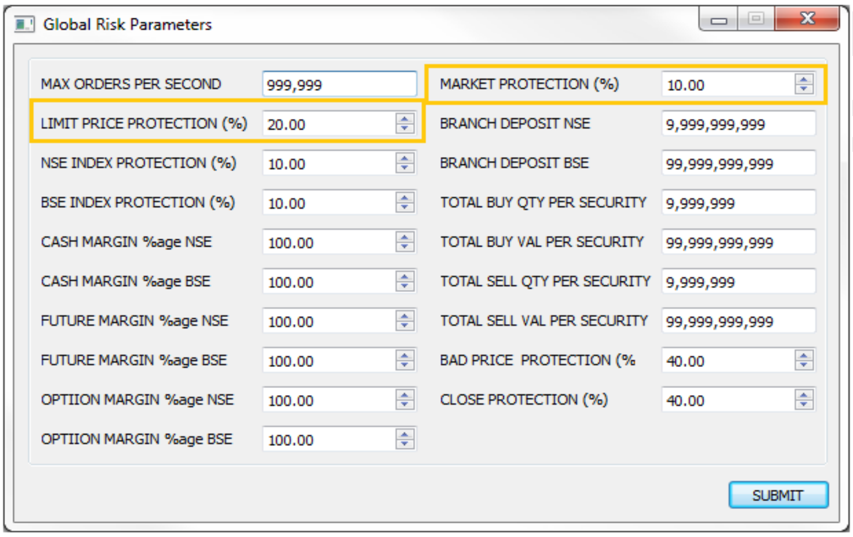


Login ID 5001

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

10.3 Price Range check

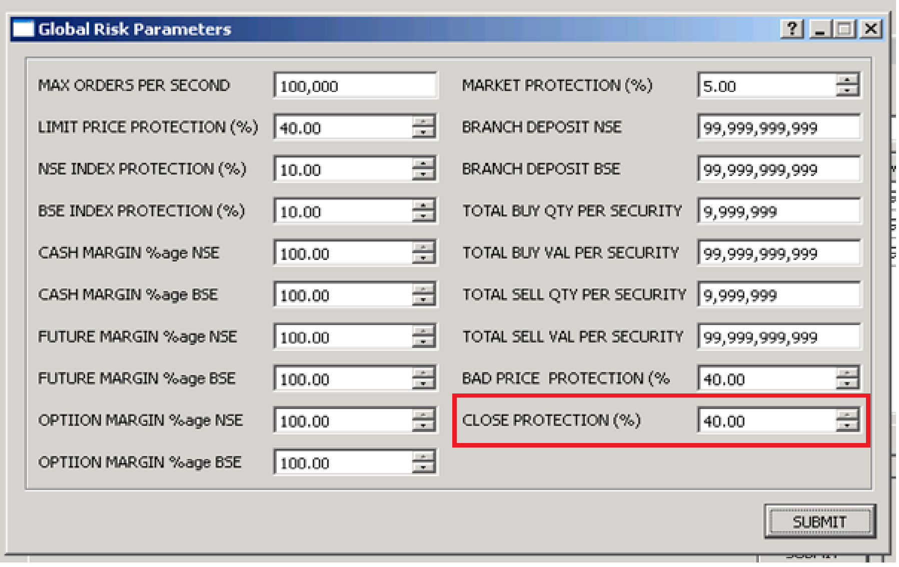
The order price generation mechanism strictly follows the specified Range provided to a trader to limit the risk within a pre-set percentage of the Last Trade Price (LTP). The percentage will be greater than zero and less than or equal to the applicable circuit limits including dummy circuit limits in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



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
OPTIION MARGIN %age NSE	100.00	CLOSE PROTECTION (%)	40.00
OPTIION MARGIN %age BSE	100.00		

10.4 Closing Price Check

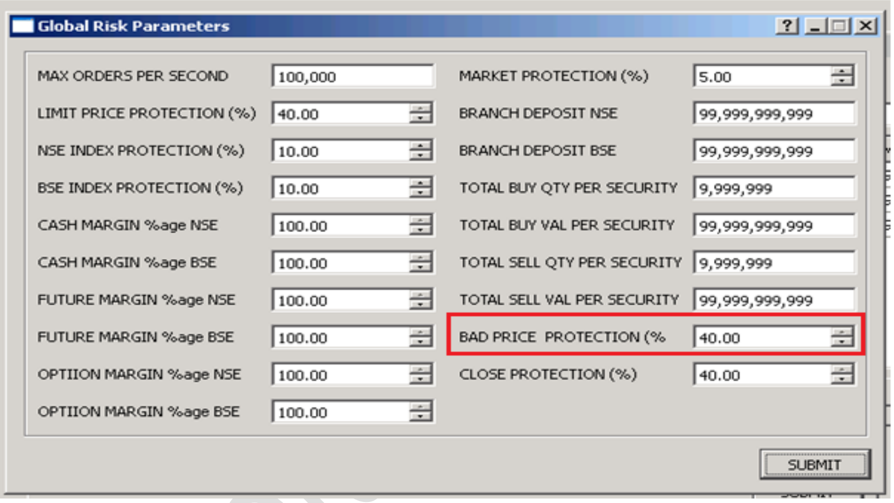
The order price generation mechanism strictly follows the specified Range provided to a trader to limit the risk within a pre-set percentage of the Closing Price.



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	BAD PRICE PROTECTION (%	40.00
OPTIION MARGIN %age NSE	100.00	CLOSE PROTECTION (%)	40.00
OPTIION MARGIN %age BSE	100.00		

10.5 Trade Price Protection Check (Bad trade Protection)

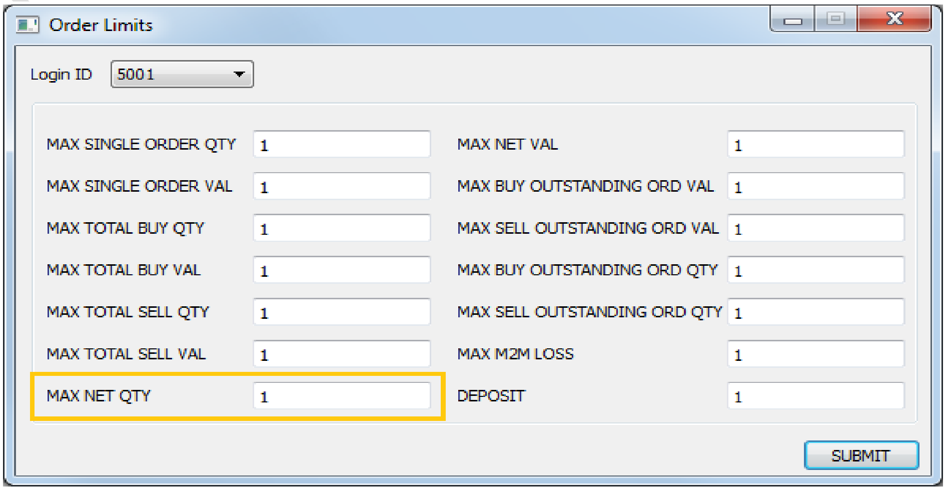
By defining a field value in Bad Price Protection % Change, order price generation can be restricted within a set percentage of last two LTP (Last trade price). Algo trade generation beyond set price bracket will result into rejection/freeze of the respective Algo rule until consecutive LTP received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



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	BAD PRICE PROTECTION (%	40.00
OPTIION MARGIN %age NSE	100.00	CLOSE PROTECTION (%)	40.00
OPTIION MARGIN %age BSE	100.00		

10.6 Position Level Check

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



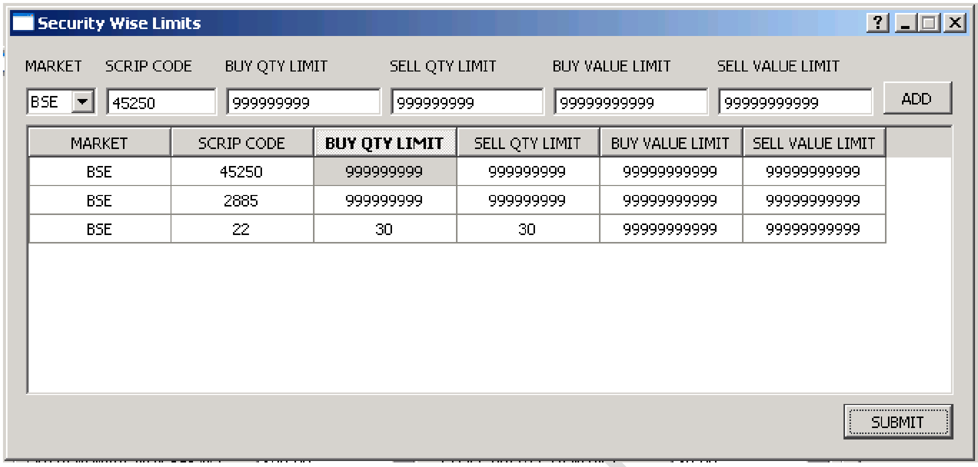
Login ID 5001

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

uTrade Sol ltd.

10.7 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.



MARKET: BSE
SCRIP CODE: 45250
BUY QTY LIMIT: 999999999
SELL QTY LIMIT: 999999999
BUY VALUE LIMIT: 99999999999
SELL VALUE LIMIT: 99999999999

MARKET	SCRIP CODE	**BUY QTY LIMIT**	SELL QTY LIMIT	BUY VALUE LIMIT	SELL VALUE LIMIT
BSE	45250	999999999	999999999	99999999999	99999999999
BSE	2885	999999999	999999999	99999999999	99999999999
BSE	22	30	30	99999999999	99999999999

11 Compliance Requirements

11.1 Order Identification

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

11.2 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.

11.3 Orders per second

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

11.4 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.

11.5 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 from to safe guard against any misuse*

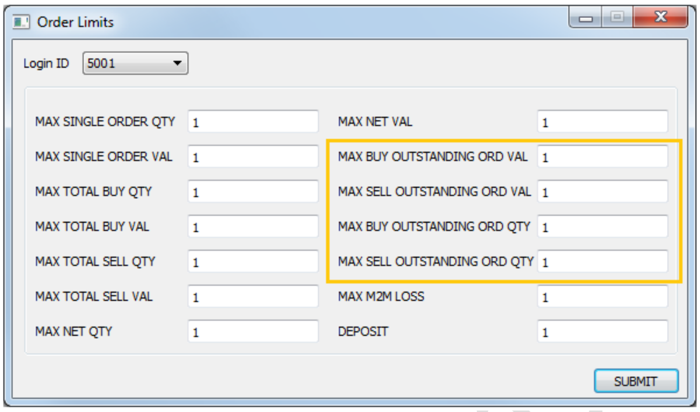
11.6 Control Values

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.

11.7 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.*

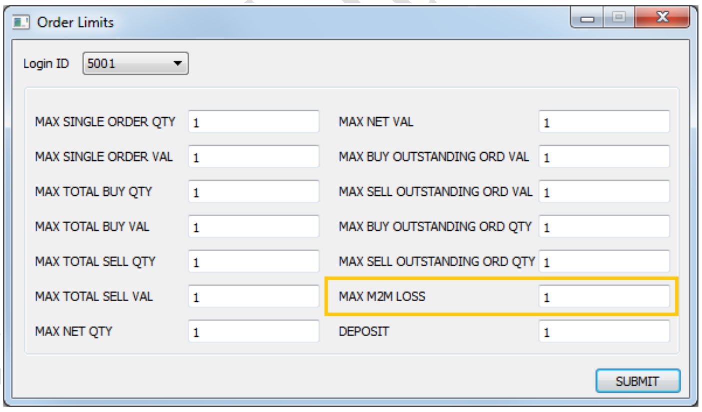


Order Limits

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

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



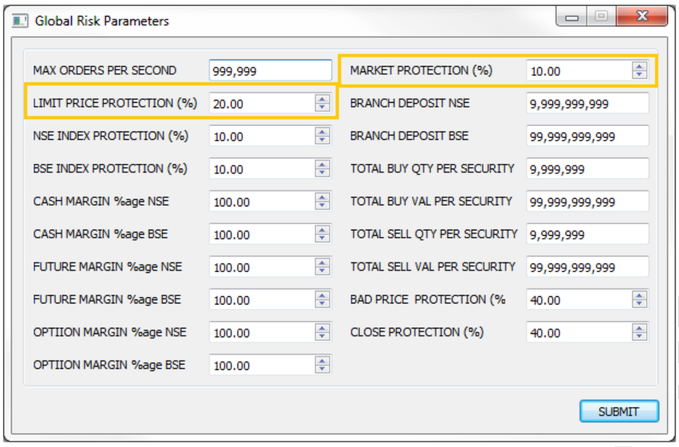
Order Limits

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

11.8 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.



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
OPTIION MARGIN %age NSE	100.00	CLOSE PROTECTION (%)	40.00
OPTIION MARGIN %age BSE	100.00		

12 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	

13 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* for further details.

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