

Efficient Strategy for Market Movement

Sonalika Papneja
M.Tech C.S.E Amity University,
Noida, India

Sanjeev Tomar
M.Tech C.S.E Amity University,
Noida, India, skumar@amity.edu

S.P. Singh
P.hd, BIT, Meshra, BIT, Noida,

ABSTRACT

Strategy is a simple computer language to automate the trading on the basis of trader's own perspective. It allows the traders to write their own strategy easily and much efficiently according to the market movements. Writing strategy in nest platform makes the user very friendly by providing the inbuilt methods which they can call whenever they need. It also includes SQL type query language to manage portfolio related queries. We will be discussing the methods and events use in implementing a strategy. We are defining the methods by giving examples along with the uses. So that traders could understand and write their own logic easily and quickly. We are also writing a sample strategy in order to get familiar with the methods provided for writing strategies. Understanding the writing of market strategy is simple but yet powerful.

GENERAL TERMS

Strategy, Token, Stocks

KEYWORDS

Order, Trade, Feed, Start up, Stop

1. INTRODUCTION

In this strategy, the required parameters from users will be buy price, sell price, buy quantity, and sell quantity and maximum quantity. In two leg order, if one leg is fired then the corresponding leg is also fired. Here the order for the tokens will be buy or sell both at once.

Inside the strategy, the market feeds are compared against the parameters given by the traders and fired orders according to the conditions written inside the strategy.

Some task are to be followed in between these strategies, subscription method should be called inside the STARTUP event, this method is use to subscribe the token id on specified fields like bestbuyprice, bestbuyquantity, bestsellprice and bestsellquantity another is getvalue which is called inside the FEED event which returns th values stored by the user.

2. PROPOSED METHODS

First, get the initial event to fire.

If the event is to **STARTUP** then follow the procedures.

Get portfolio from the portfolio universe.

- Get tokens from the portfolio.
- Retrieve the token id for the specific portfolio and the token.
- Add the token into the map by calling the function, Add To Token List Map () function.

- Mapped the tokens into a map so that later we can refer.
- Set all the values into the user defined parameters by SetReferenceParamList(Function

If the Event is **FEED**, then

- Get the token for which the event is fired.
- Count the nodes for the specified token from the map by GetListCountFromMap().
- Get the token list for the event token from the token list map.
- Next identify the types of option.
- Collect parameters from the portfolio tree.
- Collect data from market using the Get Param () method.
- After getting all the data calculate your logic and compare the strategy value with user define values.
- Check the position first, if the remaining quantity is greater than 0 then check the price.
- If the comparison is positive place the order with Place Order () function.
- If the comparison is negative wait for another tick.

If the Event is **ORDER**, then

Follow the trade event procedure

If the Event is **TRADE**, then

- Get the Token for which the event is fired.
- Get the token for which the event is fired.
- Count the nodes for the specified token from the map by GetListCountFromMap().
- Get the token list for the event token from token list map.
- Get the values for the token.
- Update the traded quantity and remaining quantity.

If the Event is **TIMER**, then

If time is over follow stop

If the EVENT is **STOP**, then

- In this event the Unsubscribe() function is called to unsubscribe all the tokens to stop trading.
-

3. ALGORITHM

3.1 Inside the STARTUP event

This is the event which fires at first whenever a strategy is run.

- The GetTime() method is called to get the fired time of event.
- Passing the query to get all the portfolios in the query type of variable i.e. qx.
- Through the GetResultSet() method, we are getting portfolio list.
 - sTime = GetTime()
 - qx = Select Portfolio from Portfoliouniverse
 - pList = GetResultSet (qx)
 - iCount= Get the Count of List (pList)
- Here a specific portfolio is getting through the GetNext() method.
- Passing the query to get specific portfolio in the query type of variable i.e. qxf.
- Getting all the tokens for the specific portfolio in the query type of variable i.e. qxtk.
 - eP1 = GetNext(pList)
 - sPortFolio = GetEntityId(eP1)
 - qxf = Select Portfolio from Portfoliouniverse in which Portfolio= sPortFolio
 - pList1 = GetResultSet(qx)
 - qxtk = Select Token from Portfoliouniverse for which Portfolio = sPortFolio
 - tokList = GetResultSet(qx)
 - iCount1 = GetListCount(tokList)
- Storing a specific token from the portfolio in a query type variable qxtk1
- Mapping the tokenId with the portfolio name and token list through the AddToTokenListMap ()method.
- Next we are setting the required fields in the portfolio tree.
 - eP11=GetNext(tokList)
 - sTokenId = GetEntityId(eP11)
 - qxtk1 = Select Token from Portfoliouniverse in which Portfolio = sPortFolio AND TOKEN = sTokenId
 - IToken = GetResultSet(qx)
 - AddToTokenListMap(sPortFolio, sTokenId, IToken)
 - e = GetNext(tokList)
 - sTokenId=GetEntityId(e)

3.2 Inside the feed event

This feed event is fired whenever price changes.

- Get the token id for which the event is fired.
- Counting the tokens from the map.
 - sEventToken=GetEventToken();
 - iListCount=GetListCountFromMap(sEventToken);
- Getting the list of the token id from the map.
- Getting the values from the portfolio tree which we had stored in the startup event.
- The GetListValue() method is converting the string value into list type.
- Here we are separating the tokens according to the option types.
- Next the GetValue() method is called
 - e = GetNext(ITokenList)
 - sTokId = GetEntityId(e)
 - sOptType = GetOptionType(sTokid)
 - if(sOptType == "CE"||sOptType == "CA")
 - sOptCallToken=sTokID
 - if(sOptType == "PE"||sOptType == "PA")
 - sOptPutToken=sTokId
 - GetValue()

4. CONCLUSION

A strategy is proposed in this paper for stock market Trading developed in context of developer and seller provided with an algorithm to explain each event of the strategy. These strategies will automate trading with the user own perspective once provided with market datas viz, BestBuyPrice, BestSellPrice, BestBuySize, BestSellSize

5. REFERENCES

- [1] Silahgi G.C and Robu V, "An agent strategy for automated stock market trading combining price and order book information" IEEE 2005 .
- [2] (2013) The NSE website. [Online] <http://www.nseindia.com/>
- [3] (2013) The BSE website. [Online] <http://www.bseindia.com/>
- [4] (2013) The NDCEX website. [Online] <http://www.ncdex.com/ding>
- [5] (2013) The MCX website. [Online] <http://www.mcxindia.com/>
- [6] Wang Yanhong and Liu Shancum, " An empirical hetrogenous trading strategy model in the shanghai stock market of china", IEEE, pp 227 – 230, 2011
- [7] Fong, s. and Tai, J., " The application of trend following strategies in stock market trading", Digital object Identifier IEEE, pp 1971 -1976, 2009
- [8] Pen-Yang Lian, Jiah-Shing Chen, " Dynamic trading strategy learning model using learning classifier systems " *IEEE* vol 2 pp 783 – 789, 2001