



Omnesys Technologies



NEST Strategy

Intraday Multi-Order Slice Jobbing

May 14, 2014

Version 1.0.0.0

Document Information

DOCUMENT CONTROL INFORMATION

| | |
|--------------------------|--|
| | |
| DOCUMENT | Nest Strategy Script Intra-day Multi-order Slice jobbing |
| VERSION | 1.0.0.0 |
| AUTHOR | RUSHABH DOSHI |
| REVIEWER | BRIJESH DAMODAR / ATHUL KUDVA |
| VERSION NOTES 1.0.0.0 | This document explains the Intraday Multi-order Slice Jobbing strategy functionality in BFO segment. |
| KEYWORDS | |

Proprietary Notice

This document contains confidential information of Omnesys Technologies Pvt Ltd, which is provided for the sole purpose of permitting the recipient to evaluate the deployment document submitted herewith. In consideration of receipt of this document, the recipient agrees to maintain such information in confidence and to not reproduce or otherwise disclose this information to any person outside the group directly responsible for evaluation of its contents, except that there is no obligation to maintain the confidentiality of any information which was known to the recipient prior to receipt of such information from Omnesys Technologies Pvt Ltd, or becomes publicly known through no fault of recipient, or is received without obligation of confidentiality from a third party owing no obligation of confidentiality to Omnesys Technologies Pvt Ltd.

Intraday Multi-Order Slice Jobbing Strategy (BFO):

This strategy allows only one BFO token (i.e. buy future scrip or sell future scrip).

Nest Strategy Client:

Portfolio Tree: Each portfolio comprises of 1 token (i.e. Future scrip 1).

Input Parameters:

Following user parameters need to be entered:

PF Name: The name of the portfolio.

Pro/Cli : It gives user an option to do in **PRO** or in **CLI**. When **CLI** is selected, it will give an option to select **Account id** and **Participant id**.

Exchange: User needs to populate relevant exchange segment. For this strategy, its BFO.

Instrument Type: User needs to select the relevant instrument type.

Scrip Name: User needs to select relevant symbol/scrip name for that particular exchange.

Expiry Date: User needs to select the relevant expiry date for the selected future expiry.

Reference Price: Price at which order is to be placed which is specified by the user.

Upper Range: It is the upper band above which the slice order will not be placed. It is specified by the user.

Lower Range: It is the lower band below which the slice order will not be placed. It is specified by the user

Buy Step: This parameter determines at which prices various buy orders should be placed from the current reference price. For example, if reference price is 5250, buy factor is Rs. - 10, and lower range is 5200, then it will place 5 buy orders at limit price 5240, 5230, 5220, 5210, and 5200. The buy orders will be placed till lower range/upper range is breached.

Sell Step: This parameter determines at which prices various sell orders should be placed from the current reference price. For example, if reference price is 5250, sell factor is Rs. 10, and upper range is 5300, then it will place 5 sell orders at limit price 5260, 5270, 5280, 5290, and 5300. The sell orders will be placed till lower range/upper range is breached.

Order type: is useful in determining whether user wants to only place (slice) orders as buy or user wants to only place (slice) orders as sell orders or user wants to place (slices) for both buy and sell orders. It should be noted even if user has selected order type as buy, it will place the reverse sell orders, once the initial buy order (slice) is executed, but it will not place initial sell orders (slices). The same logic is applicable if the order type is selected as sell.

Tick Size (Rs.): It is a minimum tick for that scrip.

Order Lots: This is the quantity (in lots) to be placed per opportunity defined for the token. The above parameters need to be updated from the front-end. Once the parameters are updated, the user needs to start the Nest Strategy from the front end so that the Nest Strategy engine starts running at backend.

Nest Strategy Engine:

The execution of the script happens in the Nest Strategy engine running at back end.

Execution:

Once the user defines the parameters such as upper range, lower range, trigger price, order type and other parameters, based on the order type selected, order price for the different slice orders on buy/sell/both sides will be placed. Order price is calculated using below-mentioned formula:

$$\text{Order price} = \text{Reference Price} + \text{Buy/Sell Step}$$

Based on the order type selected, the placement of orders will change. Strategy will work in following manner based on the order type selected as follows:

Case 1): Order type: **"Buy"**

For Example: Reference Price: 4900, Buy Step: -25 Rs., Sell Step: 10 Rs.,
Upper range: 5000, Lower range: 4800, order lots=10.

Buy Order prices will be: 4875, 4850, 4825, 4800.

Based on the above calculations, 4 buy orders will be placed at the above mentioned price for order lot 10.

Let us assume that any one of the buy order got traded say 4875 then opposite sell order will be placed with traded quantity and Price will be 1st leg placed price + sell step
i.e , 2nd leg order Price = 4875(Placed price) + 10 (Sell Step)
= 4885

Once this sell order gets completed, another buy order is placed for the price 4875.

Case 2): Order type: **"Sell"**

For Example: Reference Price: 4900, Buy Step: -10 Rs., Sell Step: 25 Rs.,
Upper range: 5000, Lower range: 4800, order lots = 10.

Sell Order prices: 4925, 4950, 4975 and 5000.

Based on the above calculations, 4 sell orders will be placed at the above mentioned price for order lot 10.

Let us assume that any one of the sell order (say 4925) got traded then opposite buy order will be placed with traded quantity and Price will be 1st leg placed price + Buy step
i.e , 2nd leg order Price = 4925(Placed price) -10 (Buy Step)
= 4915

Case 3): Order type: "Both"

For Example: Reference Price: 4900, Buy Step: -25 Rs., Sell Step: 25 Rs.,
Upper range: 5000, Lower range: 4800, order lots=10.

Buy Order Prices: 4875, 4850, 4825, 4900.

Based on the above calculations, 4 buy orders will be placed at the above mentioned price for order lot 10.

Sell Order Prices: 4925, 4950, 4975, 5000.

Based on the above calculations, 4 sell orders will be placed at the above mentioned price for order lot 10.

If any one of the order got traded on buy side or sell side then the opposite buy/sell order will be placed with the traded qty and the Price will be 1st leg placed price + Buy/sell step.

In case of partial completion of first order, the pending order is open in the market, and the opposite order is placed in the market based on the buy/sell step and the price of the first leg. As soon as anything in the second leg (opposite/reverse) order gets hit, any open orders for the initial slice is cancelled. Once the entire quantity of the second leg (Reverse/opposite) is completed, then only it will place the initial slice again. For example, let say user wants order type is sell, upper range is 5000, lower range is 4800, order lots is 10, trigger price is 4900 for a particular selected token. Based on the parameters entered, four sell orders will be placed for the price 4925, 4950, 4975, and 5000 for order lots 10 each. Let us say that 4925 got sold, but out of 10, only 4 lots got traded, hence the strategy will place opposite buy orders for $(4925 - 25 \text{ (buy step)}) = 4900$ for 4 lots, and 6 lots of sell order will be in open for price 4925. Now, at any moment, if the opposite buy order of 4900 gets traded (partially or fully), the pending 6 sell order for 4925 gets cancelled immediately. Once the entire quantity of opposite buy orders for 4900, in this case 4 lots, gets traded, system will again place a sell order for 10 lots at 4925 Rs. This will continue till the strategy is stopped.

On individual PF execution stop or global stop of strategy execution open orders gets canceled.

Portfolios and Parameters Screenshot:

MultiOrder Jobbing Strategy : 1.0.0.4

PF Name:

Exch: Inst Type: Scrip: Exp Date: PRO/CLI: Acct ID: PartID:

Reference Price: Order Type: TickSize: Upper Range: Lower Range: Buy Step: Sell Step: OrderLot:

Risk Management

Risk Management System can be configured to have following checks before the orders are released to the exchange. The checks which are defined by exchange with respect to Algorithms are in place in the system. Below mention rules can be configured in the system to control the risk parameter which is defined by exchange.

| Sr.No. | Checks | Rules to set | Remarks |
|--------|---|--|--|
| 1 | Price Check | Check Price Range Based on LTP Check Circuit Limit | This rules will create a price range on the basis of Last Traded Priced as per the percentage set in the category window. |
| 2 | Quantity Check | Order Quantity including Square off Order Board Lot Quantity including Square off Order | This rules will restrict per order the number of quantity to be placed in market which is defined the category window. The user can define the number of quantity in Weights and in lots for Futures. |
| 3 | Order Value Check | Order Value including Square off Order | This rule will restrict per order the order value which can be placed in the market which is defined in category window |
| 4 | Trade Price Protection Check | Check Circuit Limit including square off order | This rule does not allow to place the order which has been placed above the Higher Circuit Limit or Lower circuit limit which id defined for contract/scrip by exchange |
| 5 | Market Price Protection | Check Price Range Based on LTP | This rule will create a price range on the basis of Last Traded Priced as per the percentage set in the category window. |
| 6 | Cumulative Open Order Value check | Pending order value | This rule will restrict the Open Order with the Value set in the category |
| 7 | Automated Execution Check | Turnover Order Level and Turnover Order Level Limit | This rule will calculate the value of all executed/ Unexecuted and un confirm orders and if breach the value set in category then further order will get rejected |
| 8 | Automatic stoppage in event of Algo execution leading to a loop or a runaway situation. | Order Throttle | If there number of order per seconds breaches the value which is set in Throttle then further order gets rejected by the system. |
| 9 | Net Position Vs. available margin | Gross Exposure, Gross Exposure Derivative, Var Margin Order Level, Span Margin Order Level | User can set the risk parameter based on Exposure and Margin based on which the margin used will be calculated on the basis of position taken. If the Margin used is equal to Cash margin then further order |

| | | | |
|----|---|---|---|
| | | | will be rejected by the system |
| 10 | RBI Violation checks for FII Restricted stocks. | Restricted Basket or RMS Blocking | User need to create a Restricted basket for the scrip and assign to the category of the user / client. Also RMS blocking can be used. |
| 11 | MWPL violation check | RMS Ban Symbol or RMS Blocking | The scrip for which market wide position limit is breach then scrip can be blocked or it needs to be in Ban. |
| 12 | Position Limit Checks | Scrip Group / Scrip Margin | User can define the quantity scrip wise in which the position cane be taken in scrip group and then it needs to be assign to category at client level |
| 13 | Trading Limit Checks | Scrip Group / Scrip Margin | User can define the quantity scrip wise in which the position cane be taken in scrip group and then it needs to be assign to category at branch level |
| 14 | Exposure Limit check at individual client level and at overall level | Gross Exposure and Gross Exposure Limit | User can define the Exposure at Branch Level as well as Broker Level |
| 15 | Number of orders for the logic | NA | Depends on the user parameter (i.e order qty) set. This can vary from a minimum of a single lot to a maximum of total qty set. |
| 16 | Maximum number of scrips / contracts in which the logic will work at a time | NA | At a time, maximum number of scrips/contracts in which logic will work at a time is 1 |
| 17 | Number of legs | | One |