

INSTRUCTIONS FOR TOPIC 7 : CREATING ALERTS ACTIVITY

5% of your semester grade

Time required: 3 hours

Suggestion: Print this document out before trying the activity

PREPARATION

1. Create a new Microsoft Word document (use WordPad if you do not have MS Word).
2. Save your empty document to your desktop with the filename Myname_T7.doc (use your own name instead of Myname). Leave the file open while doing the activities below.

INSTRUCTIONS FOR SUBMISSION

The following activities require you to copy and paste your Alice source code into the Word document which you created above. Do NOT provide screen shots. Instead, copy and paste the source code from the ALDIT window (on the remote machine) directly into your Microsoft Word document (on your local machine). Please provide your source code for EACH of the activities below. This means you will need to provide source code for the activities 1.1, 1.2, 1.3 and 2.1. Please label them clearly in your submission.

Activity 1 (4 marks for this activity)

1.1 Please create a Short Term Price Change Alert based on the specifications given below:

Event types	Trades		
Alert Code	101		
User Parameters	Parameter	Description	Values
	SHORT_TERM_PERIOD	The number of minutes to look back	15
	BENPERIOD	The number of trading days for benchmarking	30
	ST_DEV	The number of standard deviations away from distribution mean	2
Short Text	STPM		
Long Text	SHORT TERM PRICE MOVEMENT: [security] has [increased/decreased] in price from [trueprice SHORT_TERM_PERIOD minutes ago] to [current price] in the last [SHORT_TERM_PERIOD] minutes. This is an [increase/decrease] of [price change], which is greater than the threshold of [threshold%].		
Algorithm	<p>Event 1: On every trade</p> <p>Rule 1 If the controlstatus of the security is "O" (i.e., Open)</p> <p>Determine the comparison price. If trueprice SHORT_TERM_PERIOD_101 minutes ago is defined, then set the comparison price as the trueprice SHORT_TERM_PERIOD_101 minutes ago; otherwise, set the comparison</p>		

	<p>price as the lastprice.</p> <p>Rule 2 If the comparison price is defined comparison price</p> <p>Rule 3 If the price change from the trueprice SHORT_TERM_PERIOD_101 minutes ago to the current trade price is <i>greater</i> than the threshold then</p> <p>Issue Alert 101</p> <p>Please use ALICE function trueprice(-X minutes) to get the true price SHORT_TERM_PERIOD_101 minutes ago. For a detailed explanation of true price, refer to the ALICE Reference Manual.</p>
Benchmark	<p>Benchmark Collection</p> <p>Historical distribution of SHORT_TERM_PERIOD_101 minutes price changes sampled on every on trade over the past BENPERIOD trading days.</p> <p>Threshold Determination</p> <p>Threshold = Distribution Mean + STD_DEV_101 * Distribution standard deviation</p> <p>Note: If trueprice SHORT_TERM_PERIOD_101 minutes ago is not defined, use lastprice as the comparison price.</p>
Intensity	$(\text{Price Change} - \text{Threshold}) / \text{Price Change} * 100$
Reissue	<p>101S+10</p> <p>Reissue an alert for a security if the intensity is increased by 10 between the current example, and the previous 101 alert issue (or reissue) for that security.</p>
Definitions	<p><i>price change</i> : $(\text{Price 2} - \text{Price 1}) / \text{Price 1}$</p> <p><i>Trueprice</i>: Last trade price unless the bid is now higher, or the ask is now lower. If the bid is higher, then trueprice is the bid price, if the ask is lower, then trueprice is the ask price.</p>

1.2 Please alter your solution above so that the alert will determine the approach to calculate thresholds based on the number of observations from the distribution. If the number of observations is greater than 200, use the Standard Deviation Approach; if the number of observations is greater than 50, use the distribution cutoff approach (cutoff at 95%); otherwise define a user parameter with a value of 3% and use that user parameter as the threshold.

Hints:

a) Test your script against the demo market on: 26th June, 2007

1.3 So far, the Short Term Price Change Alert you have written includes all trades regardless of whether they are on-market trades or off-market trades. Please alter your solution so that the Alert will only include on-market trades for both alerting and benchmarking.

Hints:



- a) SMARTS uses flag ON to identify on-market trades. The ALICE function `flags()` can be used to check transaction flags. For example, `flags(+ON)` will return TRUE for any on market trade; `flags(+OF)` will return true for any off-market trade.

Activity 2 (1 mark for this activity)

2.1 In SMARTS, there is a database called DAYTOT. The DAYTOT database contains daily total statistics for all securities. Some of those daily total statistics for each security are:

- Total Daily Trade Volume
- Total Daily Turnover Value
- Total Daily Trade Count
- Maximum Trade Price within the day
- Minimum Trade Price within the day
- Open Price
- Close Price

Some alerts may not need to rely on the benchmarking process we have learned in Session 7 if their thresholds only depend on statistics from the DAYTOT database. For example, an Abnormal Trade Account Alert which will be issued when the number of trades for a security on a trading day exceeds the average daily trade-count of the past 30 trading days. Please write up an ALICE script that can print out the average daily trade-count of the past 30 trading days for each security.

Hints:

- (1) Some useful functions to access the DAYTOT databases:
 - `tcount(date0, date0)` returns the total on-market volume for a security on date0.
 - `tcount(date0, date1)` returns the total on-market volume for a security between date0 and date1.
- (2) The ALICE function `trday(date0, -X)` can be used to identify the date that is X trading days prior today.
- (3) You will need to use a per security loop at start