



## INSTRUCTIONS FOR TOPIC 6 : Control Status and Loops ACTIVITY

5% of your semester grade

Time required: 2 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\_T6ACTIVITY.doc (use your own name instead of *Myname*). Leave the file open while doing the activities below.

NB The following activities require you to make screen shots of your work, and then paste these into the word document which you created above. Screenshots from the remote computer can be done by holding down ALT and then clicking the PrintScreen key on the keyboard. To quickly paste the screenshot into the word document on your local computer, just right click in the document and select paste. To compress screen shots, save them as .jpg images from within the Paint application.

### GETTING TO KNOW THE DATA

#### Introduction

Data is central to financial research and the Alice programming language is specifically designed to facilitate analysis of detailed high-frequency trading data. There is a wealth of information available in SMARTS. Examples of the data that can be extracted include; standard values such as the order price and volume of each trade; more specific variables such as broker id's or time of order entry; or calculated measures such as the length of time an order has been on the book or a stock's quoted spread. Just about every piece of information you need is available if you know where and how to look.

Much of the data in SMARTS is easy to reference while sometimes knowledge of specific flags unique to each exchange is required to access particular data fields. However, a good way to discover what data is present in the database is to try different commands and print the results to screen. This will help to determine what the dataset looks like, verify that the data is what is expected, and check for any errors. It is always a good idea to get to know the data and sort out any problems before becoming too involved in your analysis. This week's activity will show you how to get started.

#### Extracting trade data for analysis

Imagine that you are working at an exchange or a trading house and wish to take a closer look at the trading activity of a particular stock after noticing some abnormal trading behaviour. The following exercise will help you to familiarise yourself with the available data, extract specific information, and then save it to a file ready for further analysis.

## ACTIVITY 1 (2 marks)

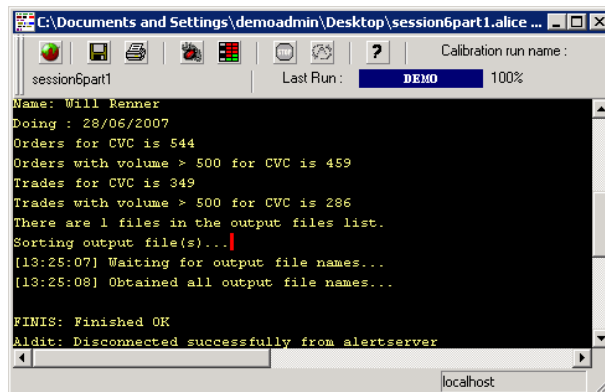
Using the Demo Market, create an Alice script that will:

- 1.1 Print out the total number of orders entered and the total number of trades for the security CVC on 28/06/2007. (0.5 marks)
- 1.2 Modify the above program (Exercise 1.1) to only print when the volume of an entered order or trade is greater than 500. (0.5 marks)
- 1.3 Now modify the above program (Exercise 1.2) so that the results will be printed to a csv file with the following columns: Date, Security, Number of Orders Entered, Number of Trades. (1 mark)

Make sure your completed program addresses all the functional requirements, then take a screenshot of both your program source code AND your program's output and paste them into your Word document.

*Hints:*

- Declare and initialize your variables first
- Use the accumulate operator ( $+=$ ) to tally the number of orders entered and trades.
- Print out the results within an *at end* code block
- The output should look like this:



```

C:\Documents and Settings\demoadmin\Desktop\session6part1.alice ...
session6part1
Name: Will Kemmer
Doing : 28/06/2007
Orders for CVC is 544
Orders with volume > 500 for CVC is 459
Trades for CVC is 349
Trades with volume > 500 for CVC is 286
There are 1 files in the output files list.
Sorting output file(s)...
[13:25:07] Waiting for output file names...
[13:25:08] Obtained all output file names...

FINIS: Finished OK
Aldit: Disconnected successfully from alertserver
localhost
  
```

Figure One

Exercise 1 requires you to count the number of orders and trades that occurred throughout the trading day and save the results to file. If you can complete this step then you have successfully processed and extracted data from SMARTS using Alice. Typically this data would then be analysed using another program such as Excel or SAS so if you have time have a go at importing the csv file you created into one of these applications.

### The state of the market

When analysing trading behaviour it is usually important to keep track of the 'state of the market' when the actions took place. For example, was trading activity highest during the opening auction, during continuous trading, or at the closing auction? Additionally, if you wish to calculate time-weighted relative spreads for example then you would probably only be interested in events that occurred during continuous trading.

In the lecture you learned that the *controlstatus* function can be used to differentiate between various market states. In this exercise we will see how to make use of this function.

## ACTIVITY 2 (3 marks)

Complete all the exercises covered in the lecture. When you are finished, build an Alice script that will:

- 2.1 Create an array to list all of the different market states encountered on 28/06/2007. (1 mark)
- 2.2 Print out the number of trades that occurred within each *controlstatus* state for all securities. (1 mark)
- 2.3 Print out the number of trades that occurred within each *controlstatus* state *per security*. (1 mark)

Make sure your completed program addresses all the functional requirements, then take a screenshot of both your program source code AND your program's output and paste them into your Word document. To get full marks for Activity 2 your program must produce the same output as Figure Three.

*Hints:*

- The *controlstatus* states you may encounter include:
  - E Pre open
  - G Opening (up)
  - O Normal trading
  - P Pre close
  - C Closed
- As a rule, lists can be incorrect, incomplete and change with time so giving your code the flexibility to handle the unexpected is good programming practice. To gain marks for Activity 2.1 your list of market states must be created dynamically rather than hard coding the set of states provided above.
- Loop over each trade and tally up the number of trades for each security AND for each controlstatus state using a two-dimensional array, e.g.:

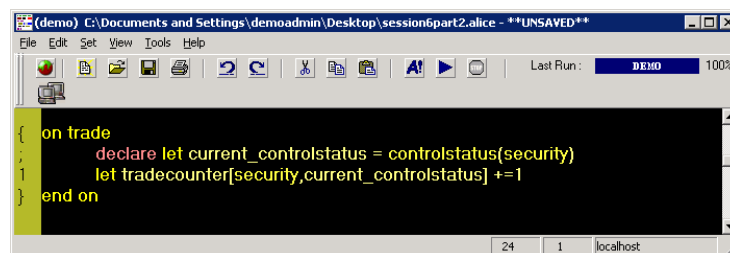


Figure Two

- Output the results by looping back over each control status state AND each security.

- The output should look like this:

```

C:\Documents and Settings\demoadmin\Desktop\session6part2.alice : session6par...
session6part2
Last Run : DEMO 100%

Name: Will Renner
Doing : 28/06/2007
---E---
MIC 3
Total trades for Control Status E is 3
---G---
CVC 14
VIS 1
Total trades for Control Status G is 15
---O---
CPI 104
CVC 332
FFB 10
ICO 202
IFI 33
localhost
    
```

Figure Three

In this activity you learned how to separate trading activity depending on the state of the market. In future you will frequently find use for the *controlstatus* function which we have covered here.

### SUMMARY

This week's activities focussed on getting you to know the data. You should now have some idea how to:

- Query the SMARTS data and extract basic information such as order price and volume for securities of interest
- Carry out basic processing of this data such as filtering out only large trades
- Save extracted data to file
- Determine the state of the market when an order or trade occurred

In the next lesson we will interrogate the data more thoroughly so that we do not need to extract it. We will look at how to identify interesting features in the data such as abnormal end-of-day price movements and automatically detect them by creating our first alert using Alice.

### SUBMISSION

- Check your Word document and make sure it contains four screen shots, two for each of the activities above. The screen shots for Activity One should be on page one. The screen shots for Activity Two should be on page two.
- Make sure your word document is named in this way: MYNAME\_T6ACTIVITY.doc (use your own name instead of MYNAME).
- In the output examples above, replace *Will Renner* with your own name.
- Submit your document into Canvas by uploading it on the assignment submissions page.