Capital Markets CRC Limited

MARKET DATA ANALYSIS USING ALICE

Controlstatus and Loops

Controlstatus and Loops

Presented by
Professor Michael Aitken
Chief Scientist, CMCRC
Chairman, Smarts Group

Prepared by
Shan Ji & Will Renner

Controlstatus and Loops

In this lecture
- *Controlstatus*
- Defined and undefined
- Per loops
- Printing to csv files

Time
- 35 Minutes

Requirements
- Complete all activities in Session 5

Control status

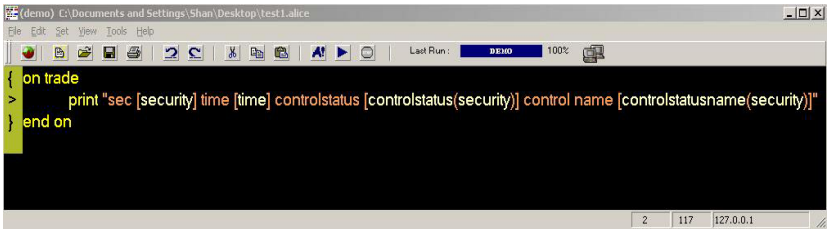
Control Status: an event in the order book database
- On control: a *when clause* for control status
- Functions to obtain controlstatus:
  - Controlstatus(security)
  - Controlstatusname(security)
  - Used within a *transaction* when clause block

Control status

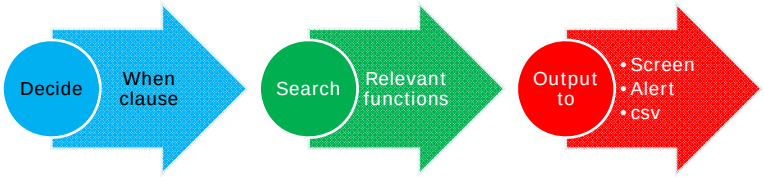
- Example

Control status

- Example



- A simple ALICE programming workflow:



Control status

- Execute the ALICE script
- Are you happy with the results?
- Is the script printing too much to the screen?

IF Statement

How do we use *If statements*?

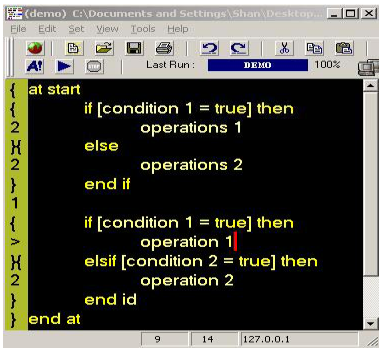
- A filter: so we can extract only those transactions of interest.

IF statements - syntax

If [true] then … end if

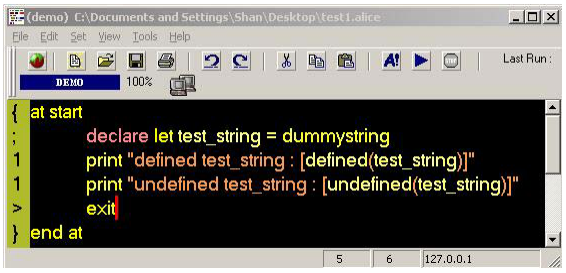
if [true] then … else … end if

If [true] then … elsif [true] then … end if



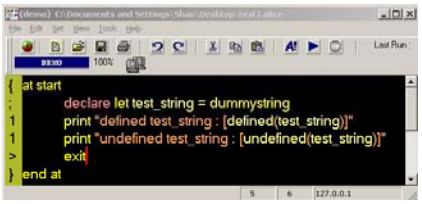
IF statements

- Let's see how we can use an If Statement to make the previous *Controlstatus* script only print out the possible values of controlstatus with which a trade can occur
- First, let's learn the "defined" & "undefined" conditions.

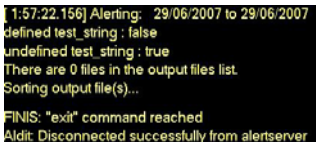


Guess what the output will be?

Defined and undefined



Source code



Output

Before you assign any values to a variable or when you assign a *dummy value* to a variable the variable's value is *undefined.*

- The Defined() and Undefined() functions check if a variable's value is defined or not

Dummy values & boolean variables

- Setting undefined values in ALICE:
  - dummyprice, dummyvolume, dummysecurity, etc
- Boolean variables
  - Should have two states: TRUE and FALSE
  - Actually, there are three possible states for a boolean variable: TRUE, FALSE and UNDEFINED

Assume: TestString = dummystring

- True eg. undefined(TestString)
- False eg. defined(TestString)
- Undefined eg. print TestString

IF Statement

- if defined(MyVariable) then …
- if defined(MyVariable) = true then …

- If undefined(MyVariable) then …
- If undefined(MyVariable) = true then …

Recap: Other conditional operators:

- If variable1 **=** variable2 then …
- If variable1 **>** variable2 then …
- If variable1 **<** variable2 then …
- If variable1 **!=** variable2 then…
- If variable1 **>=** variable2 then…
- If variable1 **<=** variable2 then…

NOTE: **!=** Stands for 'not equal' to

Exercise 1

- Use an *if statement* to print
  - "undefined" when Test_Security is undefined;
  - "defined" when Test_Security is defined;
- Hint: use the *if… then…else* structure.

```
at start
        declare let Test_Security = ^CVC
        if defined(Test_Security) then
```

PAUSE HERE

Exercise 1 - solution

```
at start
        declare let Test_Security = ^CVC
        if defined(Test_Security) then
                print "defined"
        else
                print "undefined"
        end if
        exit
end at
```

Exercise 1 - solution

```
at start
        declare let Test_Security = ^CVC
        if defined(Test_Security) then
                print "defined"
        else
                print "undefined"
        end if
        exit
end at
```

```
at start
        declare let Test_Security = ^CVC
        if defined(Test_Security) then
                print "defined"
        elsif undefined(Test_Security)
                print "undefined"
        end if
        exit
end at
```

IF Statement

- Now let's go back to the task raised before:
  - use If Condition to make the previous "Controlstatus" script only print out possible controlstatus at which a trade can occur.
- To solve this task, we will need to use
  - If statement
  - Boolean: undefined()
  - Array: Controlstatus_Seen[controlstatus]
- The key is not to print controlstatus that have been seen already

Exercise 2 – fill in the blank

Part 1:Have a look at the following partial solution for the "controlstatus" task and see if you can fill the red part

```
declare Controlstatus_Seen[string] : boolean

on trade
        declare let current_controlstatus = controlstatus(security)
        if undefined(Controlstatus_Seen[current_controlstatus]) then
                print "[security] [current_controlstatus] [controlstatusname(security)]"
                Controlstatus_Seen[                    ] =
        end if
end on
```

PAUSE HERE

Exercise 2

- Solution

```
declare Controlstatus_Seen[string] : boolean

on trade
        declare let current_controlstatus = controlstatus(security)
        if undefined(Controlstatus_Seen[current_controlstatus]) then
                print "[security] [current_controlstatus] [controlstatusname(security)]"
                Controlstatus_Seen[current_controlstatus] = true
        end if
end on
```

- Compare your solution with the one shown

Exercise 2

- Part 2: Now please execute the solution for Exercise 2 on your computer.
- Expected Output

```
Doing : 29/06/2007
C Closed
O Open
G Opening
There are 0 files in the output files list.
Sorting output file(s)...

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

- Question: What types of trades can occur when the market is Closed, Open and Opening?

Exercise 2

What types of trades can occur when the market is:

- Closed (controlstatus = C)
- Opening (Controlstatus = G)?

Refer to the lecture for Session 2: *Understanding the Trading Market*.

- *Off Market Trades* can take place when the market is closed
- *Opening Auction Trades* occur when the market is opening

If Statements – Multiple Conditions

- Single condition: To print out details of trade for security CVC

```
on trade
        if security = ^CVC then
                print "time [time] sec [security] price [price] volume [volume]"
        end if
end on
```

- Use “and”, “or” to join multiple conditions
- To process only those orders entered for security CVC before 14:00:00

```
if security=CVC AND time<14:00 then…
    <put some code in here>
end if
```

If Statements – Multiple Conditions

Print the total number of orders entered for security CVC before 14:00:00

- Use AND or OR to join multiple conditions
- "+=" stands for accumulation
- "+=1" means increase the value of NumOfOrders by 1

```
declare NumOfOrders : number
on entord
        if security = ^CVC and time < 14:00:00 then
                NumOfOrders += 1
        end if
end on

at end
        print "Num Of Ordres entered: [NumOfOrders]"
end at
```

Loops

- To repeat a particular operation
- In ALICE, there are two types of loops
  - Per Loop
    Per security, Per house, Per Trader, etc
  - For Loop

Per loop

- *Per loops* loop over an entity
- eg. per security loops over all securities in the market; per house loops over all brokers in the market

```
at start
        per security
                print security
        end per
end at
```

```
[ 3:54:43.562] Alerting:   29/06/2007 to 29/06/2007
^CPI
^CVC
^FFB
^HPI
^ICO
^IFI
^IIC
```

Exercise 3

Use *per loops* to print out the long name of all securities as well as the long name of all brokers.

Hints:

- to retrieve the security long name, you will need to use the function *securityfield(security, "LN", date)*
- to retrieve broker long name, you will need to use the function *housefield(house, "LN", date)*

PAUSE HERE

Exercise 3

- One Solution is:

```
at start
        per security
                print "security [security] FULL NAME [securityfield(security, "LN", date)]"
        end per

        per house
                print "Broker [house] FULL NAME [housefield(house, "LN", date)]"
        end per
end at
```

- Right-click the *securityfield* for adhoc help.

For loops

- Unlike the PER LOOP, FOR LOOPs allow us to loop a limited number of times.
- Syntax:

for [i = initial value]; [condition]; [loop step] do

Operations go in here!

end for

```
declare sum : number
at start
        for declare let i = 1; i <= 100; i += 1 do
                sum += i
        end for
        print "sum 1 to 100 = [sum]"
        exit
end at
```

Exercise 4

- Stock markets are open for trading on trading days. They normally close on public-holidays. How many trading days are there between 01/01/2007 and 29/06/2007?
- Hint: In ALICE, the function istrading(date) can be used to check if a day is a trading day.

PAUSE HERE

Exercise 4

- Solution

```
declare NumOfTrdays : number
at start
        declare let start_date = 01/01/2007
        declare let end_date = 29/06/2007
        for declare let _date = start_date; _date <= end_date; _date += 1 do
                if istrading(_date) then
                        NumOfTrdays += 1
                end if
        end for
        print "Number Of Trading Days = [NumOfTrdays]"
end at
```

Printcsv

- So far, we have always displayed outputs from our ALICE programs in the Run Log Window by using the “print” function
- ALICE can print to csv as well.
- Sytax:
  printcsv “[filename.csv]”, col1, col2, etc.

Printcsv

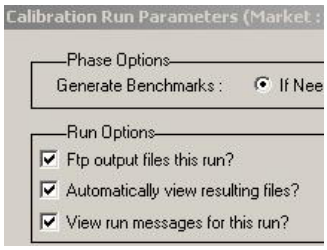
- Print the following details of every trade which occurred on 29/06/2007 to a csv file:
  - Security
  - Trade Date
  - Trade Time
  - Trade Price
  - Trade Volume
  - Trade Value

Printcsv

- Example

```
at start
        printcsv "TradeDetails.csv", "Security", "Trade Date",
                                     "Trade Time", "Price",
                                     "Volume", "Value"
end at

on trade
        printcsv "TradeDetails.csv", security, date,
                                     time, price,
                                     volume, value
end on
```



When you run the above script, please select:

- **Ftp Output files this run** and
- **Automatically view resulting files**

as shown above

Key terms and concepts

- Controlstatus
- If statements
- Booleans
- Undefined and Defined
- True or False
- Per loops
- For loops
- printcsv

Help is available

- Review this lecture
- Consult the Alice Reference Manual
- Post a question to the discussion board