HOOTIE

Highly Organized & Optimized Trading In Exchanges

Houtan M. Fanisalek
HMF2117@columbia.edu

Introduction

Most traders have little or no technical knowledge and rely on software and human interaction to perform complex trading routines. The primary objective of this language is to provide a way for traders to create an automated black box system that is simple enough to understand and use yet robust enough for high volume trading. The language and compiler will handle all of the details leaving the user free from worrying about actual code.

The Language

The HOOTIE language is very simple and those that have the basic knowledge of trading stocks can pick it up very quickly. Each command is written on a single line with no semicolon and each item is separated by spaces. The symbol is in reference to a stock symbol.

Commenting: // COMMENT //	
Looking up a Stock: LOOKUP *Symbol [Information]*	[Information] VOLUME BETA MCAP RANGE
When trading with stocks: BUY *Symbol Quantity [Conditions] [Term]* Example: BUY MGM 2000 [LIMIT 15.00] [30DAYS] SELL *Symbol Quantity [Conditions] [Term]* Example: SELL AIG 2000 [LIMIT 34.00] [60DAYS] **When trading with options:** Buy2Open *Symbol Quantity [Conditions] [Term]* Buy2Close *Symbol Quantity [Conditions] [Term]* Sell2Open *Symbol Quantity [Conditions] [Term]* Sell2Close *Symbol Quantity [Conditions] [Term]* *Note that the commands above will be queued until the market allows for the buy or sell to complete.*	[Conditions] MARKET (this is the default) LIMIT *value* STOP *value* STOPLIMIT *value* TRADESTOP$ *value* TRADESTOP% *value* [Term] NOW (default) TODAY 7DAYS 30DAYS 60DAYS 120DAYS

Logic Keywords & Control Statements:

WHEN (Symbol IS Condition) { }
OTHERWISE { }
LOOP (Symbol IS Condition) { }
WAIT *time(ms)*

Wait for a Buy or Sell to Execute
WAITEXE

To exit a control block
BREAKOUT

Comparisons
AND
OR
NOT
GREATER
LESS
GREATERORE
LESSORE

Jump to a block
GOTO *blockname*

[Condition]

VOLUME *value*
BETA *value*
MCAP *value*
RANGE *value1 value2*

Variables
SET @variablename
UNSET @variablename

Example:
@NumberOfTrades = 0
@NumberOfTrades = @NumberOfTrades + 1

Output:
PRINT *command*
PRINT *string*

Example:
PRINT “STOCK IS IN RANGE!”

Special Blocks:
Each execute block is done simultaneously like in VHDL and keeps looping.
EXECUTE *blockname*:

To exit a execute block
STOP

To exit the program completely
EXIT

Example:

```
EXECUTE a_merger {
        WHEN (BLOAQ IS GREATORE .009) {
                SELL NFLX 2000
                STOP
        }
}
```

Exceptions: **BAD_SYMBOL** **NOT_IN_PORTFOLIO** **NOT_ENOUGH_MONEY** **MARGIN_CALLED** **EPIC_FAILURE**	

Code Sample

```
PRINT LOOKUP NFLX
PRINT LOOKUP BLOAQ

EXECUTE wait_for_range {
        WHEN (NFLX IS GREATER 210.0) {
                PRINT “NETFLIX IS TOO EXPENSIVE!”
                EXIT
        }
}

EXECUTE wait_for_volume {
        WHEN (NFLX IS VOLUME GREATER 10MIL) {
                SELL NFLX 2000
                WAITEXE
                EXIT
        }
}

EXECUTE a_merger {
        WHEN (BLOAQ IS GREATORE .009) {
                SELL NFLX 2000
                STOP
        }
}
```