(J)Robotrader Manual 0.2.8

Table of Contents

Objectives

The objective of the robotrader is to create a simple simulated automated trading system using technical indicators, machine learning methods and other well-known trading strategies. The scope is currently limited to trading methods that base their decisions only on historical pricing information. The pricing information is extracted from Yahoo! Finance but it is our intention to include other data providers in the future.

Chapter 1. Introduction

1. Intended Audience

The robotrader is a java application that can be used by any investor who is interested in the results of technical trading methods.

Chapter 2. Installation

Download the latest version from SourceForge. Extract the archive file to your favourite directory. Edit the configuration files as required. Start either the GUI or the command line tool.

1. Configuration files

The traders xml file configures the automated traders that are competing against each other. The current default traders file can be found under `conf/traders.xml`. The first line of the file denotes the version of the xml and the encoding of the characters as follows

```
<?xml version="1.0" encoding="UTF-8"?>
```

The `<traders> </traders>` xml tags indicate the start and the end of the trader xml elements. Each of the traders is denoted with the `<trader class="[full java class name]"/>` with the class attribute pointing to the actual java class name. This suffices to include the trader within the list of traders that are to compete against each other. However, in many cases further properties can be set to configure some of the parameters of each trader, such as the length of the period of historic prices to include for the decision making. These properties are added as follows

```
<property key="[key]" value="[value]"/>
```

Chapter 3. Simple Automated Traders

The traders below follow simple trading strategies

1. Keeper (`robotrader.trader.example.Keeper`)

The Keeper is the simplest of all traders. It invests all of its funds in the stock at the start of the trading period and keeps (hence Keeper) the acquired positions until the end of the trading period. The Keeper is a copy of the performance of the underlying instrument. Any other technical trader should beat the Keeper trader in order to beat the market.

2. Follower (`robotrader.trader.example.Follower`)

The Follower, as the name suggests, follows the streaks of the underlying instrument as long as the price is non-decreasing. Whenever the price goes down, the positions in the stock are sold.

3. Win Follower (`robotrader.trader.example.WinFollower`)

The WinFollower invests all the funds in the instrument when the price goes up and keeps the positions until a minimum percentage of profit is made. In that case, the stock is sold. The same situation arises when a maximum loss percentage is encountered as all positions are sold in this case as well. The default percentages are 10% for minimum profit and 5% for maximum loss.

4. Streak Follower (`robotrader.trader.example.StreakFollower`)

The StreakFollower buys or sells all its position in the investment whenever the direction of the streak remains constant for a minimum number of streaks. The trader therefore buys the stock when a minimum number of streaks with increasing prices are encountered and sells the positions when a minimum number of streaks with decreasing prices for the stock are found. The default number of up streaks is 3 and down streaks is 0.

5. Cautious Streak Follower (`robotrader.trader.example.CautiousStreakFollower`)

The CautiousStreakFollower is similar to the StreakFollower, however, instead of spending all its cash on the stock when the minimum number of up streaks is encountered, only a percentage of the cash is spend. The default percentage is 50%.

6. Turning Point Trader (`robotrader.trader.example.TurningPointTrader`)

The Turning Point Trader tries to determine the turning points of a time series by very simple means. It employs three levels

- Buying Level: Is a certain percentage above the minimum price encountered during a period
- Selling Level 1: Is a certain percentage below the maximum price encountered over the period
- Selling Level 2: Is a certain percentage above the buy level

The trader buys the shares of the instrument whenever the current price is above the buy level but below both sell levels. When the price falls below the first sell level and the buy level is below that sell level, the stock is sold. Furthermore, if the price is above the second sell level, the stock is also sold.

7. Cost Average Trader (`robotrader.trader.example.TurningPointTrader`)

The cost average trader spreads its funds evenly over the entire trading period in constant intervals. It is supposed to resemble the trader that invests the same amount of cash into the same stock each week/

month/ quarter or year.

Chapter 4. Technical Indicators

The following traders are based on popular technical indicators.

1. DeMarker (`robotrader.trader.indicator.DeMarkerTrader`)

Demarker Technical Indicator is based on the comparison of the period maximum with the previous period maximum. If the current period (bar) maximum is higher, the respective difference between the two will be registered. If the current maximum is lower or equaling the maximum of the previous period, the naught value will be registered. The differences received for N periods are then summarized. The received value is used as the numerator of the DeMarker and will be divided by the same value plus the sum of differences between the price minima of the previous and the current periods (bars).

If the current price minimum is greater than that of the previous bar, the naught value will be registered. When the indicator falls below 30, the bullish price reversal should be expected. When the indicator rises above 70, the bearish price reversal should be expected. If you use periods of longer duration, when calculating the indicator, you'll be able to catch the long term market tendency. Indicators based on short periods let you enter the market at the point of the least risk and plan the time of transaction so that it falls in with the major trend.

2. Moving Average Convergence Divergence (MACD) (`robotrader.trader.indicator.MACDTrader`)

Moving Average Convergence/Divergence is the next trend-following dynamic indicator. It indicates the correlation between two price moving averages. The Moving Average Convergence/Divergence Technical Indicator is the difference between a 26-period and 12-period Exponential Moving Average (EMA).

In order to clearly show buy/sell opportunities, a so-called signal line (9-period indicators` moving average) is plotted on the MACD chart. The MACD proves most effective in wide-swinging trading markets.

There are three popular ways to use the Moving Average Convergence/Divergence:

- crossovers
- overbought/oversold conditions, and
- divergences

The basic MACD trading rule is to sell when the MACD falls below its signal line. Similarly, a buy signal occurs when the Moving Average Convergence/Divergence rises above its signal line. It is also popular to buy/sell when the MACD goes above/below zero.

The MACD is also useful as an overbought/oversold indicator. When the shorter moving average pulls away dramatically from the longer moving average (i.e., the MACD rises), it is likely that the security price is overextending and will soon return to more realistic levels.

An indication that an end to the current trend may be near occurs when the MACD diverges from the security. A bullish divergence occurs when the Moving Average Convergence/Divergence indicator is making new highs while prices fail to reach new highs. A bearish divergence occurs when the MACD is making new lows while prices fail to reach new lows. Both of these divergences are most significant when they occur at relatively overbought/oversold levels.

3. Money Flow Index (`robotrader.trader.indicator.MFITrader`)

The Money Flow Index (MFI) is a momentum indicator that is similar to the Relative Strength Index (RSI) in both interpretation and calculation. However, MFI is a more rigid indicator in that it is volume-weighted, and is therefore a good measure of the strength of money flowing in and out of a security. It compares "positive money flow" to "negative money flow" to create an indicator that can be compared to price in order to identify the strength or weakness of a trend. Like the RSI, the MFI is measured on a 0 - 100 scale and is often calculated using a 14 day period. The "flow" of money is the product of price and volume and shows the demand for a security and a certain price. The money flow is not the same as the Money Flow Index but rather is a component of calculating it. So when calculating the money flow, we first need to find the average price for a period. Since we are often looking at a 14-day period, we will calculate the typical price for a day and use that to create a 14-day average.

$$\text{Typical Price} = \frac{\text{Day High} + \text{Day Low} + \text{Day Close}}{3}$$

Money Flow = (Typical Price) x (Volume) The MFI compares the ratio of "positive" money flow and "negative" money flow. If typical price today is greater than yesterday, it is considered positive money. For a 14-day average, the sum of all positive money for those 14 days is the positive money flow. The MFI is based on the ratio of positive/negative money flow (Money Ratio).

$$\text{Money Ratio} = \frac{\text{Positive Money Flow}}{\text{Negative Money Flow}}$$

$$\text{Money Flow Index} = 100 - \left[\frac{100}{1 + \text{Money Ratio}}\right]$$

The MFI can be interpreted much like the RSI in that it can signal divergences and overbought/oversold conditions. Positive and negative divergences between the stock and the MFI can be used as buy and sell signals respectively, for they often indicate the imminent reversal of a trend. If the stock price is falling, but positive money flow tends to be greater than negative money flow, then there is more volume associated with daily price rises than with the price drops. This suggests a weak downtrend that threatens to reverse as money flowing into the security is "stronger" than money flowing out of it. As with the RSI, the MFI can be used to determine if there is too much or too little volume associated with a security. A stock is considered "overbought" if the MFI indicator reaches 80 and above (a bearish reading). On the other end of the spectrum, a bullish reading of 20 and below suggests a stock is "oversold".

4. Relative Strength Index (`robotrader.trader.indicator.RSITrader`)

Developed by J. Welles Wilder and introduced in his 1978 book, New Concepts in Technical Trading Systems, the Relative Strength Index (RSI) is an extremely useful and popular momentum oscillator. The RSI compares the magnitude of a stock's recent gains to the magnitude of its recent losses and turns that information into a number that ranges from 0 to 100. It takes a single parameter, the number of time periods to use in the calculation. In his book, Wilder recommends using 14 periods. The RSI's full name is actually rather unfortunate as it is easily confused with other forms of Relative Strength analysis such as John Murphy's "Relative Strength" charts and IBD's "Relative Strength" rankings. Most other kinds of "Relative Strength" stuff involve using more than one stock in the calculation. Like most true indicators, the RSI only needs one stock to be computed. In order to avoid confusion, many people avoid using the RSI's full name and just call it "the RSI."

$$\text{RSI} = 100 - \frac{100}{1 + \text{RS}}$$

$$\text{Average Gain} = \underline{(\text{Total Gains}/n)}$$
$$\text{Average Loss} = (\text{Total Losses}/n)$$

$$\text{First RS} = (\text{Average Gain}/\text{Average Loss})$$

$$\text{Smoothed RS} = \frac{[(\text{previous Average Gain}) \times 13 + \text{Current Gain}]/14}{[(\text{previous Average Loss}) \times 13 + \text{Current Loss}]/14}$$

$$n = \text{number of RSI periods}$$

Chapter 5. Machine Learning Methods

These traders that employ machine learning algorithms for time-series forecasting.

1. Regression (`robotrader.trader.machinelearning.RegressionTrader`)

The linear regression trader is based on the single variable linear regression forecasting model from Openforecast. The trader tries to fit the training data to a straight line by minimising the mean sum of the squared errors between the line and the data.

2. Polynomial Regression (`robotrader.trader.machinelearning.PolynomialRegressionTrader`)

The linear regression trader is based on the single variable ploynomial regression forecasting model from Openforecast. Works in a similar fashion to the regression trader, however, it tries to fit the training data to a polynomial function.

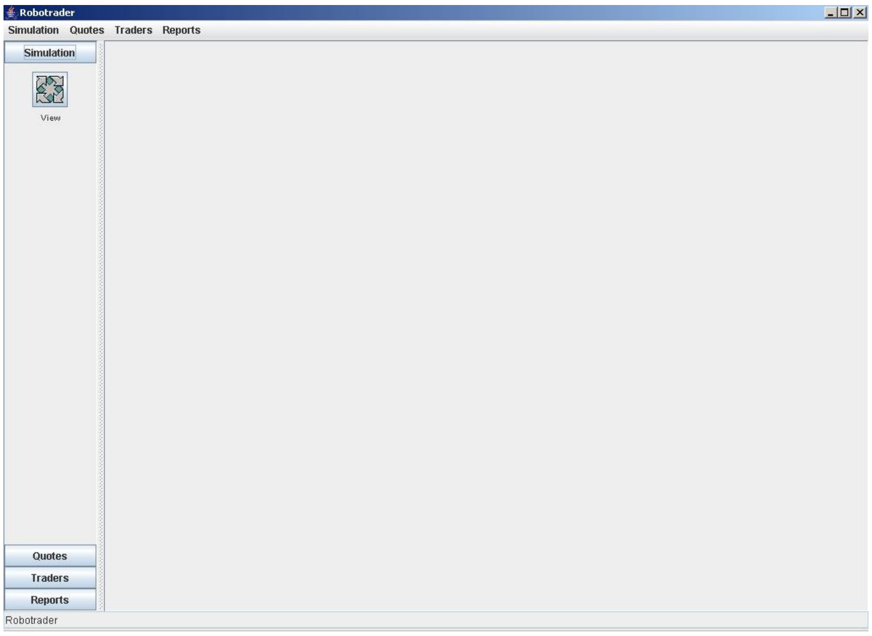
3. Neural Network (`robotrader.trader.machinelearning.JooneNeuralNetworkTrader`)

The neural network forecasting trader is based on the Java Object-Oriented Neural Network Engine Joone. The architecture can be summarised as follows: The training data is normalised and then fed into a delay layer which stores up to `taps+1` previous prices. These prices are then forwarded to the sigmoid layers of the neural network. The teacher compares the output of the network with the calculated turning points and the errors are fed back to the network to facilitate learning. A turning point exists whenever a minimum change percentage is encountered for the price time series. The learning process is repeated `epoch` times.

Chapter 6. Main Graphical User Interface

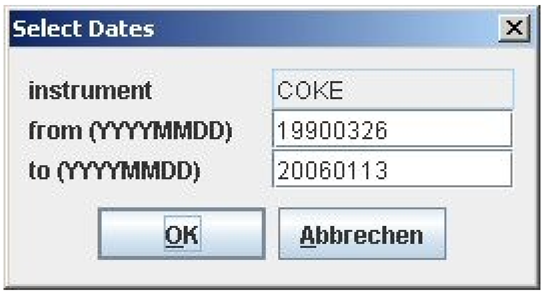
When you run the startGui.bat, the following window appears. You can either scroll the buttons on the left-hand side which work in a similar fashion to MS Outlook. Alternatively, you can choose your selection from the menu bar at the top.

Figure 6.1. The Start Window



When you click on the Simulation -> View button (or select the menu) you get a window similar to the one below. The quote file specifies the ticker used by Yahoo finance. In this instance, the COKE is modelled. The trader file denotes the location of the xml file which specifies all the traders to include together with their parameters.

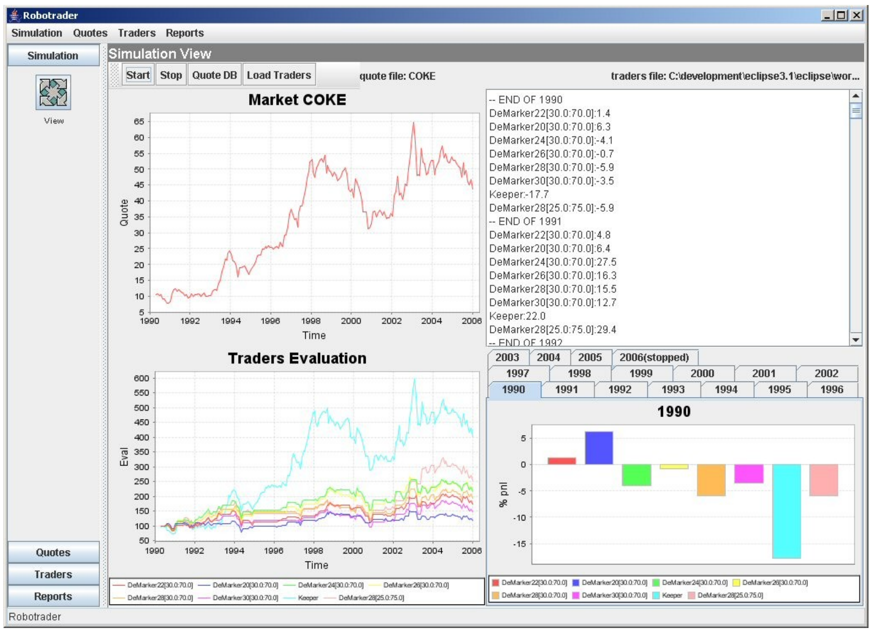
Figure 6.2. The Dates Dialog



Select Dates

instrument	COKE
from (YYYYMMDD)	19900326
to (YYYYMMDD)	20060113

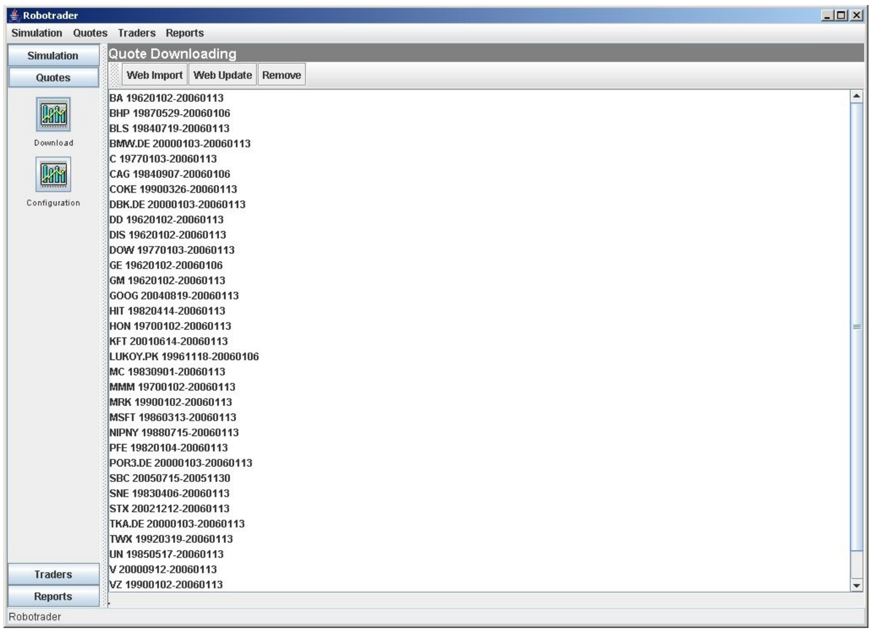
Figure 6.3. The View Window (for COKE)



Robotrader
Simulation Quotes Traders Reports
Simulation
View
Simulation View
Start
Stop
Quote DB
Load Traders
quote file: COKE
traders file: C:\development\eclipse3.1\eclipse\wor...
Market COKE
Quote
Time
Traders Evaluation
Eval
Time
DeMarker22[30.0:70.0] DeMarker20[30.0:70.0] DeMarker24[30.0:70.0] DeMarker26[30.0:70.0]
DeMarker28[30.0:70.0] DeMarker30[30.0:70.0] Keeper DeMarker28[25.0:75.0]
2003 2004 2005 2006(stopped)
1997 1998 1999 2000 2001 2002
1990 1991 1992 1993 1994 1995 1996
1990
% pnl
Quotes
Traders
Reports
Robotrader

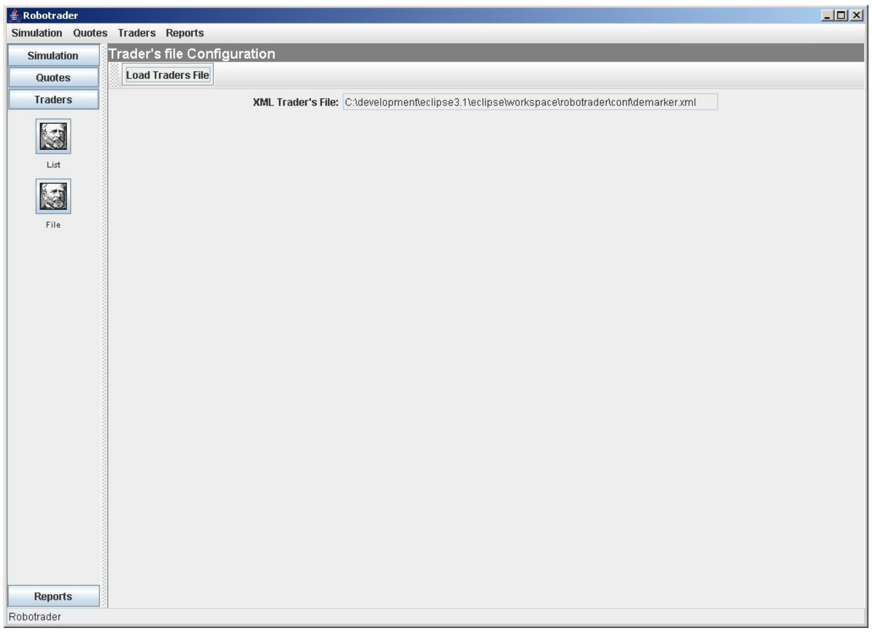
```
-- END OF 1990
DeMarker22[30.0:70.0]:1.4
DeMarker20[30.0:70.0]:6.3
DeMarker24[30.0:70.0]:-4.1
DeMarker26[30.0:70.0]:-0.7
DeMarker28[30.0:70.0]:-5.9
DeMarker30[30.0:70.0]:-3.5
Keeper:-17.7
DeMarker28[25.0:75.0]:-5.9
-- END OF 1991
DeMarker22[30.0:70.0]:4.8
DeMarker20[30.0:70.0]:6.4
DeMarker24[30.0:70.0]:27.5
DeMarker26[30.0:70.0]:16.3
DeMarker28[30.0:70.0]:15.5
DeMarker30[30.0:70.0]:12.7
Keeper:22.0
DeMarker28[25.0:75.0]:29.4
-- END OF 1992
```

Figure 6.4. The Quotes Panel



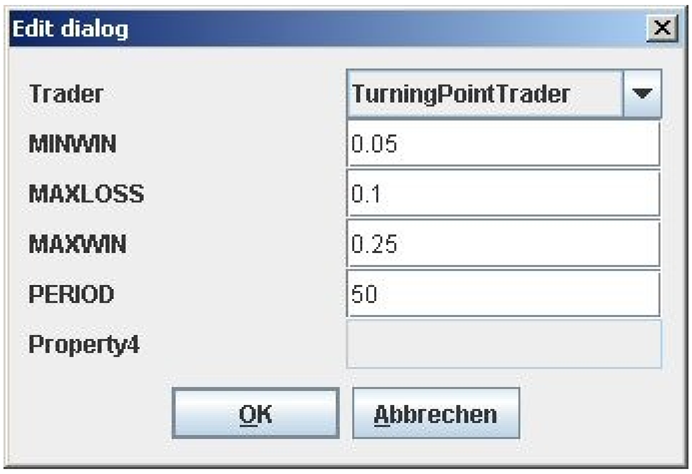
```
BA 19620102-20060113
BHP 19870529-20060106
BLS 19840719-20060113
BMW.DE 20000103-20060113
C 19770103-20060113
CAG 19840907-20060106
COKE 19900326-20060113
DBK.DE 20000103-20060113
DD 19620102-20060113
DIS 19620102-20060113
DOW 19770103-20060113
GE 19620102-20060106
GM 19620102-20060113
GOOG 20040819-20060113
HIT 19820414-20060113
HON 19700102-20060113
KFT 20010614-20060113
LUKOY.PK 19961118-20060106
MC 19830901-20060113
MMM 19700102-20060113
MRK 19900102-20060113
MSFT 19860313-20060113
NIPNY 19880715-20060113
PFE 19820104-20060113
POR3.DE 20000103-20060113
SBC 20050715-20051130
SNE 19830406-20060113
STX 20021212-20060113
TKA.DE 20000103-20060113
TWX 19920319-20060113
UN 19850517-20060113
V 20000912-20060113
VZ 19900102-20060113
```

Figure 6.5. Load Traders Panel



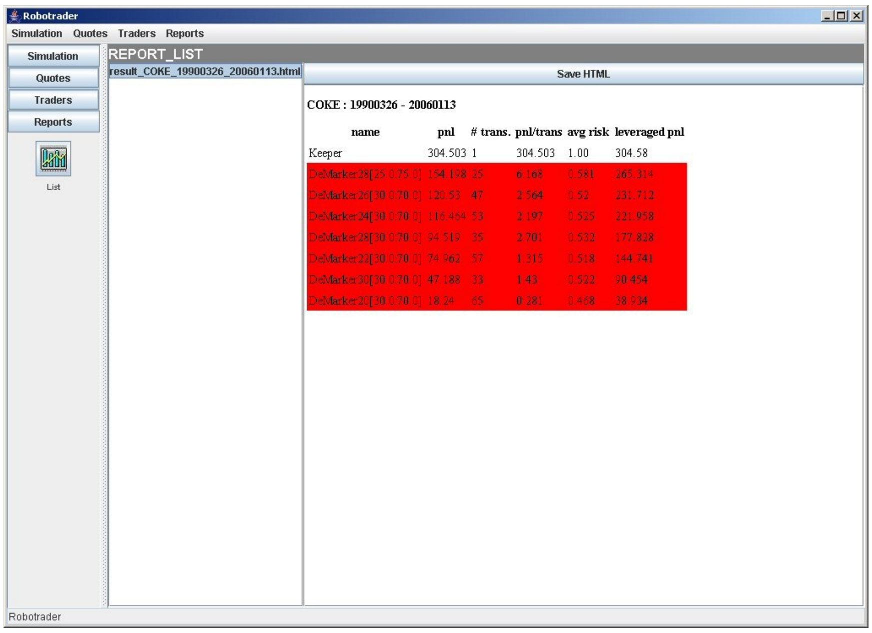
XML Trader's File: C:\development\eclipse3.1\eclipse\workspace\robotrader\conf\demarker.xml

Figure 6.6. Edit Traders Dialog



Trader	TurningPointTrader
MINWIN	0.05
MAXLOSS	0.1
MAXWIN	0.25
PERIOD	50
Property4	

Figure 6.7. Report Panel



result_COKE_19900326_20060113.html

COKE : 19900326 - 20060113

name	pnl	# trans.	pnl/trans	avg risk	leveraged pnl
Keeper	304.503	1	304.503	1.00	304.58
DeMarker28[25 0.75 0]	154.198	25	6.168	0.581	265.314
DeMarker26[30 0.70 0]	120.53	47	2.564	0.52	231.712
DeMarker24[30 0.70 0]	116.464	53	2.197	0.525	221.958
DeMarker28[30 0.70 0]	94.519	35	2.701	0.532	177.828
DeMarker22[30 0.70 0]	74.962	57	1.315	0.518	144.741
DeMarker30[30 0.70 0]	47.188	33	1.43	0.522	90.454
DeMarker20[30 0.70 0]	18.24	65	0.281	0.468	38.934

Chapter 7. Examples of Trading Performances

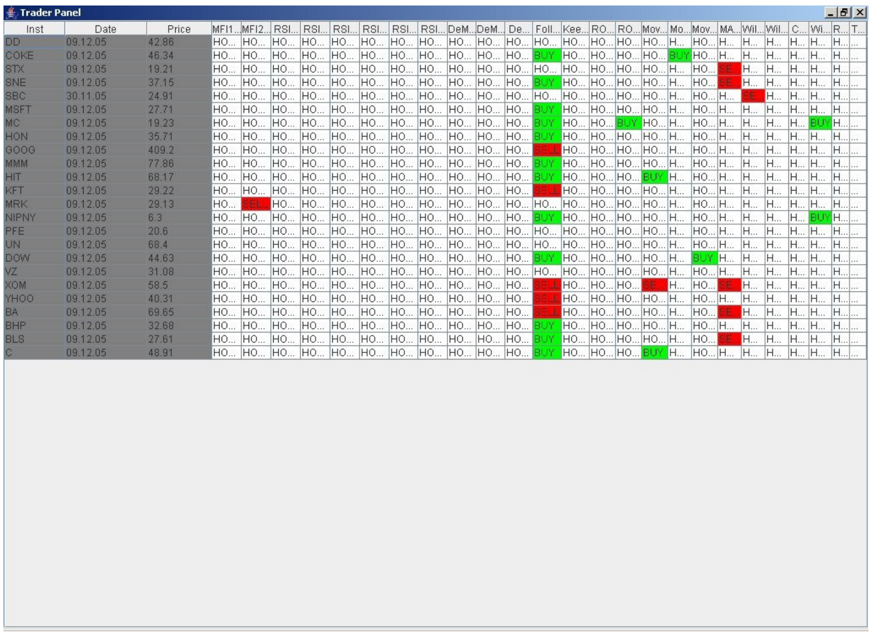
To be done.

Chapter 8. Indicator Panel

The indicator panel displays the latest trading decisions of the traders. It requires three parameters, the path where all the tickers can be found (e.g. conf/), the path to the master quote file (e.g. conf/quotes.xml) and the path to the traders (e.g. conf/traders.xml).

```
java robotrader.indicator.gui.IndicatorPanelMain tickerspath
masterquotefilepath tradersfilepath
```

Figure 8.1. Indicator Panels



Trader Panel

Inst	Date	Price	MFI1...	MFI2...	RSI...	RSI...	RSI...	RSI...	RSI...	RSI...	DeM...	DeM...	De...	Foll...	Kee...	RO...	RO...	Mov...	Mo...	Mov...	MA...	Wil...	Wil...	C...	Wi...	R...	T...
DD	09.12.05	42.86	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	H...	HO...	H...	H...	H...	H...	H...	H...	...
COKE	09.12.05	46.34	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	BUY	HO...	HO...	HO...	HO...	BUY	HO...	H...	H...	H...	H...	H...	H...	...
STX	09.12.05	19.21	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	H...	HO...	SE...	H...	H...	H...	H...	H...	...
SNE	09.12.05	37.15	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	BUY	HO...	HO...	HO...	HO...	H...	HO...	SE...	H...	H...	H...	H...	H...	...
SBC	30.11.05	24.91	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	H...	HO...	H...	SE...	H...	H...	H...	H...	...
MSFT	09.12.05	27.71	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	BUY	HO...	HO...	HO...	HO...	H...	HO...	H...	H...	H...	H...	H...	H...	...
MC	09.12.05	19.23	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	BUY	HO...	HO...	BUY	HO...	H...	HO...	H...	H...	H...	H...	BUY	H...	...
HON	09.12.05	35.71	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	BUY	HO...	HO...	HO...	HO...	H...	HO...	H...	H...	H...	H...	H...	H...	...
GOOG	09.12.05	409.2	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	SELL	HO...	HO...	HO...	HO...	H...	HO...	H...	H...	H...	H...	H...	H...	...
MMM	09.12.05	77.86	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	BUY	HO...	HO...	HO...	HO...	H...	HO...	H...	H...	H...	H...	H...	H...	...
HIT	09.12.05	68.17	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	BUY	HO...	HO...	HO...	BUY	H...	HO...	H...	H...	H...	H...	H...	H...	...
KFT	09.12.05	29.22	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	SELL	HO...	HO...	HO...	HO...	H...	HO...	H...	H...	H...	H...	H...	H...	...
MRK	09.12.05	29.13	HO...	SEL...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	H...	HO...	H...	H...	H...	H...	H...	H...	...
NIPNY	09.12.05	6.3	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	BUY	HO...	HO...	HO...	HO...	H...	HO...	H...	H...	H...	H...	BUY	H...	...
PFE	09.12.05	20.6	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	H...	HO...	H...	H...	H...	H...	H...	H...	...
UN	09.12.05	68.4	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	H...	HO...	H...	H...	H...	H...	H...	H...	...
DOW	09.12.05	44.63	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	BUY	HO...	HO...	HO...	HO...	H...	BUY	H...	H...	H...	H...	H...	H...	...
VZ	09.12.05	31.08	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	H...	HO...	H...	H...	H...	H...	H...	H...	...
XOM	09.12.05	58.5	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	SELL	HO...	HO...	HO...	SE...	H...	HO...	SE...	H...	H...	H...	H...	H...	...
YHOO	09.12.05	40.31	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	SELL	HO...	HO...	HO...	HO...	H...	HO...	H...	H...	H...	H...	H...	H...	...
BA	09.12.05	69.65	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	SELL	HO...	HO...	HO...	HO...	H...	HO...	SE...	H...	H...	H...	H...	H...	...
BHP	09.12.05	32.68	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	BUY	HO...	HO...	HO...	HO...	H...	HO...	H...	H...	H...	H...	H...	H...	...
BLS	09.12.05	27.61	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	BUY	HO...	HO...	HO...	HO...	H...	HO...	SE...	H...	H...	H...	H...	H...	...
C	09.12.05	48.91	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	HO...	BUY	HO...	HO...	HO...	BUY	H...	HO...	H...	H...	H...	H...	H...	H...	...

Chapter 9. Command Line Tools

There are two command line tools that facilitate the batch processing of the automated traders as well as automatic price updates from Yahoo! Finance.

1. Main Console

The main console runs the trading algorithms included in a trader configuration file against either a single instrument or all instruments in batch mode. It requires three command line options that need to be given prior to execution.

```
java robotrader.ConsoleMain xmlmasterquotefile xmltraderfile
instrument|-all
```

2. Automatic Price Updates

The automatic price updater takes a list of tickers and the master quote file name and updates the prices of the instruments until the most recent available at Yahoo! Finance.

```
java robotrader.quotedb.web.UpdateQuotesMain tickersfilepath
xmlmasterquotefile [loadfrom US(default)|FR]
```