Omnesys Technologies Pvt. Ltd.

NEST PLUS

NEST Pulse User Manual

September, 2011

Document Information

DOCUMENT CONTROL INFORMATION	
DOCUMENT	Nest Pulse User Manual
VERSION	1.0.0.0
VERSION NOTES	This document explains about the features and functionalities of Nest Pulse product
KEYWORDS	

What is NEST Pulse?

NEST Pulse is a trading tool that allows a trader to trade based on technical analysis using advanced charts and over 40 technical indicators. It allows the user to write his own strategies, back-test them and on satisfactory results, execute them in the live market.

NEST Pulse makes it easy for the user to not just identify opportunities based on several technical indicators, but also trade on those opportunities. In case the user is not comfortable with taking a given strategy to the live market, backtesting will help in studying how the strategy would have worked on historical feeds for that day, whether the strategy would have been profitable or not, and the efficiency of trades.

Getting Started with NEST Pulse

Once you have subscribed to the NEST Pulse service, you can start using it by logging into your NEST Trader application and Launching Nest Plus.

The first step to start using NEST Pulse is to invoke charts. This can be done by selecting any given scrip in the Market Watch and using the shortcut key Shift+P. An intraday chart as illustrated below would get displayed.

How to use Charts?

Charts are graphical representation of market data such as price and volume of stocks and other financial instruments plotted sequentially over time.

Charts help you in studying price movements and apply various technical indicators at the same time to analyze price data for a given stock. Such data can be viewed and analyzed in many different ways, depending on how the chart is plotted.

- Chart Styles: Chart styles define how the chart must be plotted for any given data. Following are the chart styles that the user can use to view charts in NEST Pulse.

Line Chart – A line chart displays information as a series of data points connected by straight line segments.

Bar Chart – A bar chart is a chart with rectangular bars with lengths proportional to the values that they represent.

Candle Chart – A candlestick chart is a combination of a line-chart and a bar-chart, in that each bar represents the range of price movement over a given time interval.

Standard – A standard chart is an OHLC chart that represents price movement for a financial instrument over a given time interval, such that it uses vertical lines for High and Low and horizontal intercepts for Open and Close prices.

Standard HLC – This chart style is similar to the Standard chart style, except that with this type of chart, Open price is disregarded and only High, Low and Close are illustrated.

- Price Styles: Price Styles define what data is used and how this data is used to plot charts. Following are the price styles that the user can use to view charts in Nest Pulse.

Standard – A standard price style simply uses market data for price and volume, in relation to the time interval selected to plot a chart.

Point Figure – Point and figure chart style is used in technical analysis, to attempt to predict financial market prices. This type of chart does not plot price against time, instead it plots price against changes in direction by plotting a column of Xs as the price rises and a column of Os as the price falls.

Renko – Renko is a type of chart that is only concerned with price movement; time and volume are not included. A renko chart is constructed by placing a brick in the next column once the price surpasses the top or bottom of the previous brick by a predefined amount. This type of chart is supposed to be effective for traders to identify key support/resistance levels.

Kagi – The Kagi chart style is used for tracking price movements and to make decisions on purchasing stock.It uses a series of vertical lines to illustrate general levels of supply and demand for certain assets. This type of chart is independent of time and only changes direction once a predefined reversal amount is reached.

Three line Break – A Three Line Break chart consists of vertical lines (boxes) connected to each other. The direction of the lines is drawn based on price movements. With this type of chart, the time relation, trading volume and chart information such as high and low prices are not taken into account. Three Line Break charts are especially useful for detecting changes in trends.

Equivolume – The equivolume chart style is a price plot that incorporates volume into each period. An Equivolume box consists of three components: price high (forming the upper boundary), price low (forming the lower boundary) and volume (depicted by the width). This makes it easier to verify volume for reversals, big moves, support/resistance breaks and climaxes.

Candle volume – A candle volume chart style combines the features of candle stick and equivolume charts. With this type of chart, the open and close during an interval form the upper and lower boundary for the chart, the high and low are marked by the wicks or lines similar to a candle stick, and the volume is depicted by the width of the chart.

Heiken-Ashi – The Heikin-Ashi chart is constructed like a regular candlestick chart, except that it plots the Close as the average price of the current bar and the Open as the mid-point of the previous bar. It is one of many techniques used in conjunction with candlestick charts to improve the isolation of trends and to predict future prices.

Technical Indicators:

A technical indicator is a series of data points that are derived by applying a formula to the price data of a security. Price data includes any combination of the open, high, low or close over a period of time. Some indicators may use only the closing prices, while others incorporate volume and open interest into their formulas. The price data is entered into the formula and a data point is produced.

A technical indicator offers a different angle from which to analyze the price action and can provide unique perspective on the strength and direction of the underlying price action.

Double-click on any of the indicators that you wish to apply to a chart.

Following are the indicators that you can use in NEST Pulse to not just study and analyze price data, but also act on them:

- Accumulative Swing Index
- Aroon
- Aroon Oscillator
- Bollinger Bands

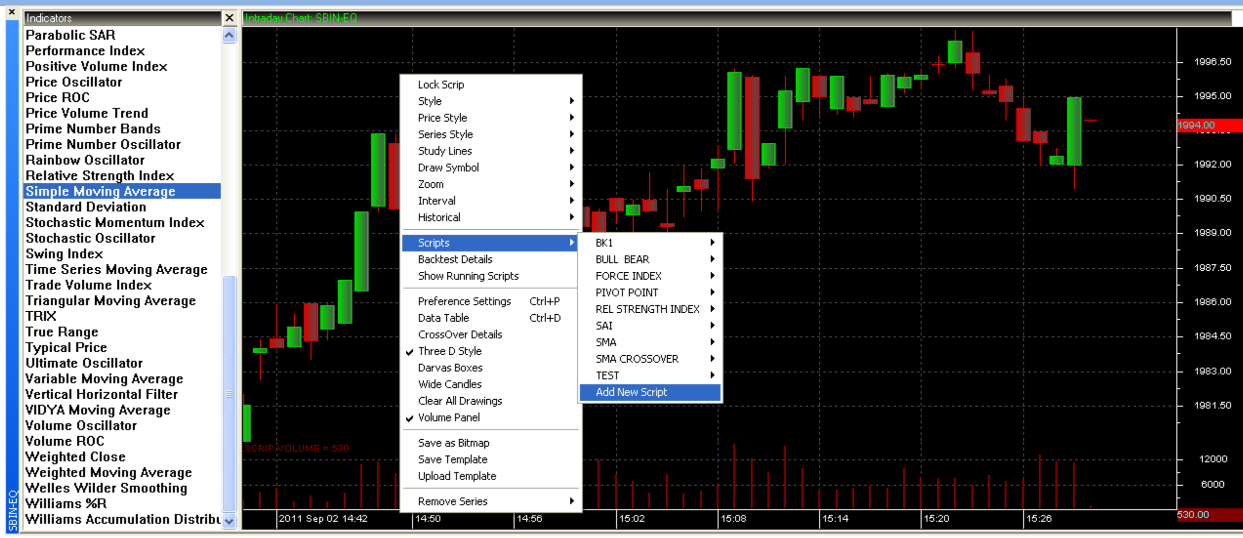
- Chaikin Money Flow
- Chaikin Volatility
- Chande Momentum Oscillator
- Commodity Channel Index
- Comparative RSI
- Detrended Price Oscillator
- Directional Movement System
- Ease of Movement
- Exponential Moving Average
- Fractal Chaos Bands
- Fractal Chaos Oscillator
- High Minus Low
- High/ Low Bands
- Historical Volatility
- Linear Regression Forecast
- Linear Regression Intercept
- Linear Regression R-Squared
- Linear Regression Slope
- MACD
- MACD Histogram
- Mass Index
- Median Price
- Momentum Oscillator
- Money Flow Index
- Moving Average Envelope
- Negative Volume Index

- On Balance Volume
- Parabolic SAR
- Performance Index
- Positive Volume Index
- Price Oscillator
- Price ROC
- Price Volume Trend
- Prime Number Bands
- Prime Number Oscillator
- Rainbow Oscillator
- Relative Strength Index
- Simple Moving Average
- Standard Deviation
- Stochastic Momentum Index
- Stochastic Oscillator
- Swing Index
- Time Series Moving Average
- Trade Volume Index
- Triangular Moving Average
- TRIX
- True Range
- Typical Price
- Ultimate Oscillator
- Variable Moving Average
- Vertical Horizontal Filter
- VIDYA Moving Average

- Volume Oscillator
- Volume ROC
- Weighted Close
- Weighted Moving Average
- Welles Wilder Smoothing
- Williams %R
- Williams Accumulation Distribution

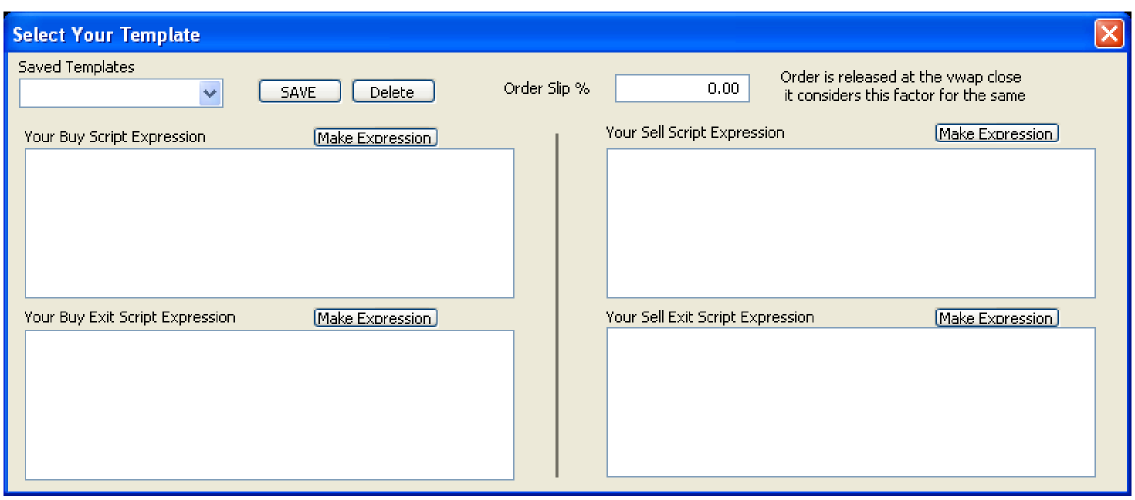
Write your own Strategy (Scripts):

Using NEST Pulse, you can create your own strategies using the technical indicators discussed above for buying/ selling an instrument. Right-click on a chart → Scripts → Add New Script, to write your strategy.

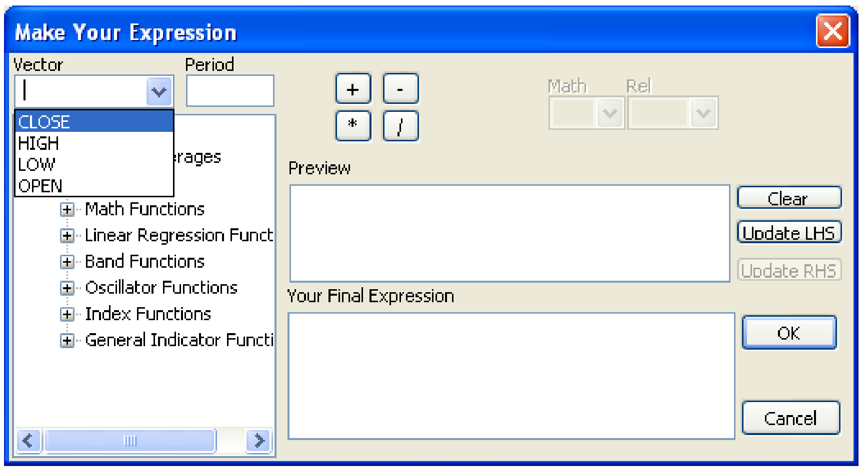


A strategy can be written with four expressions:

1. Buy Script Expression – to define when a long (buy) position should be taken for a scrip.
2. Sell Script Expression – to define when a short (sell) position should be taken for a scrip.
3. Buy Exit Script Expression – to define when to exit a long (buy) position by selling that position.
4. Sell Exit Script Expression – to define when to exit a short (sell) position by buying that position.



This can be done either by entering the strategy using the edit boxes provided or by using the 'Make Expression' function.



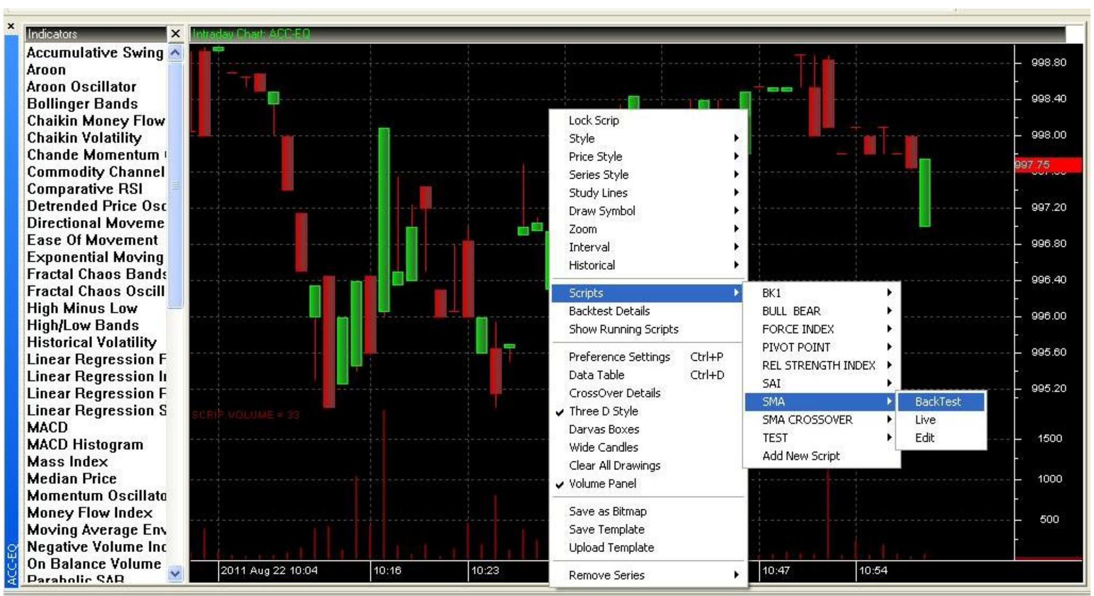
Select the required parameters for Vector/ Period/ Indicator and click on Update LHS to update the left-hand side of the final equation. Select a math function and again the desired parameters to complete the expression by clicking on Update RHS. To set additional conditions like 'AND', 'OR' and so on, you can select the condition under 'Rel' drop-down. Select the parameters for 'Vector', 'Period' and 'Function' and click on Update RHS to complete your expression.

In case you do not wish to set a condition for all expressions, you would have to type in 0 in the edit boxes for those expressions. Only the expression for which you set any condition would generate orders, provided that condition is met. Save your template by entering any name in the Saved Templates option and click on Save. You're now ready to backtest, go live or edit your strategy.

Back-testing:

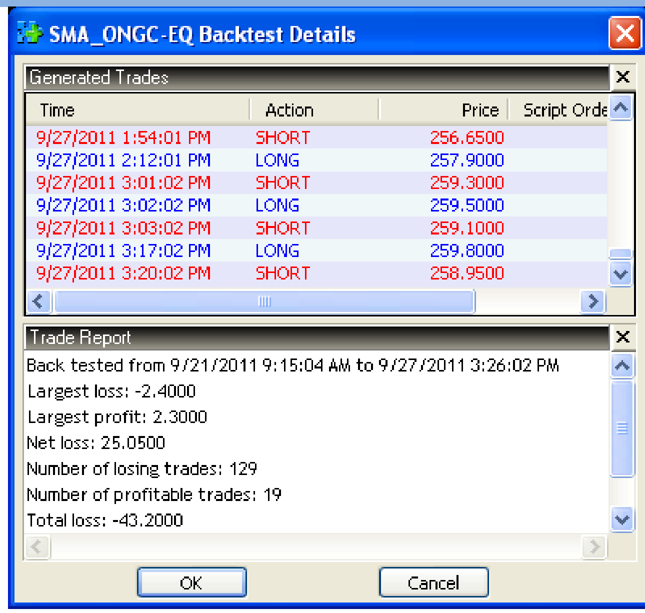
Back-testing helps you to analyze his strategy and test it out by applying it on historical data before going live with it. If you're not satisfied with the outcome as indicated by the backtest for a particular strategy, you can modify the expression for that strategy and backtest it again until the results appear satisfactory.

To back-test a strategy, right-click on the chart, click on Scripts, select the script name that you want to back-test and click on Backtest.



It will indicate buy/ sell signals generated (if any) for that script in the chart window. An upward green arrow indicates a Buy signal, while a downward red arrow indicates a Sell signal.

The user can also view comprehensive details of the Back Test, by right-clicking on the Intraday Chart and selecting ‘Back Test Details’.The Back Test Details window, as illustrated below, provides details with respect to the outcome of the strategy entered, evaluated on historical data. Corresponding to the user’s parameters getting triggered the action is taken and profit/ loss calculated, which the user can use as a reference to check whether the back-test was successful, or he would like to use a different methodology.



The Backtest Trade Report section provides details such as the Total number of trades, number of profitable and loss-making trades, largest profit/ loss, net profit/ loss and so on to better help in analysing the applicability of a strategy.

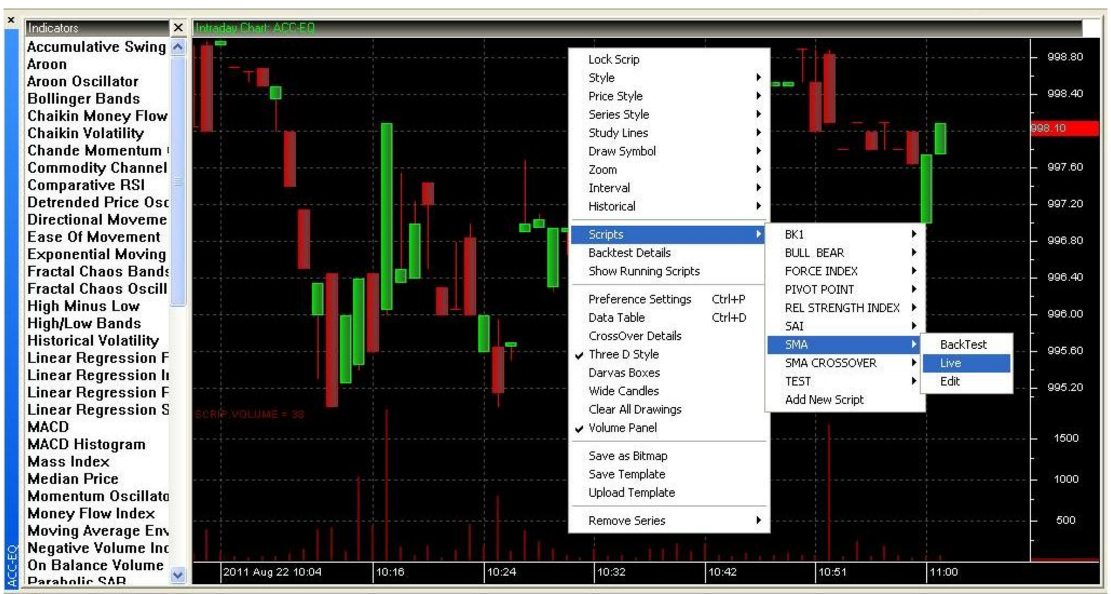
Going Live with your Strategy (Live Script):

'Going live with your strategy' means that whenever your condition as set in the Buy, Sell, Buy Exit, Sell Exit expressions is met, corresponding orders will start getting placed in the Live Market.

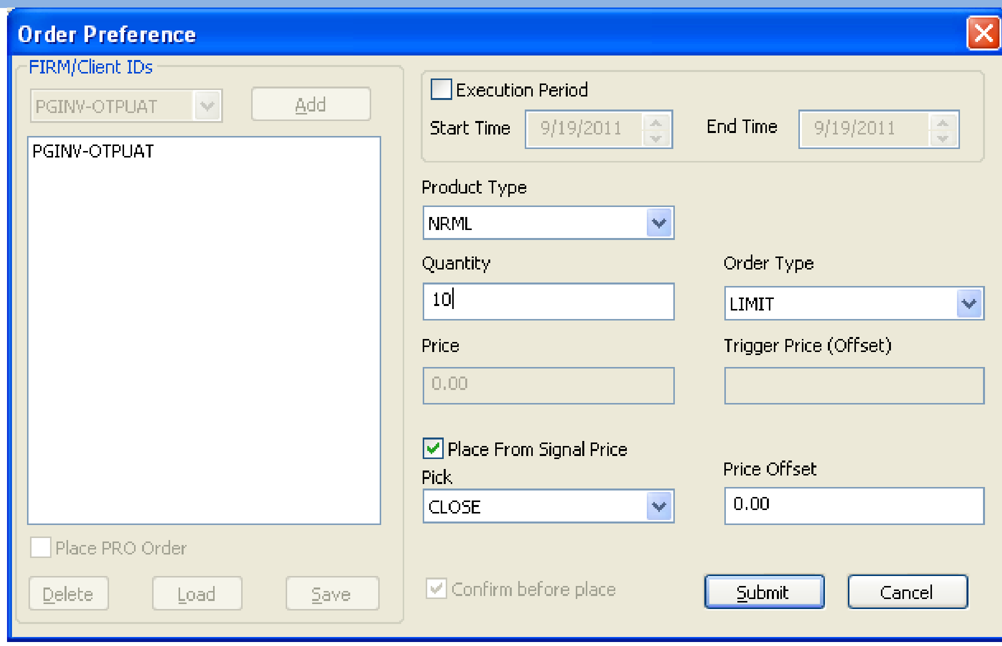
In case you're using Nest Pulse with an Investor Client login, you will be prompted to confirm the order, every time a condition is triggered. Unless you confirm the order to get placed, it will not.

In case you're using Nest Pulse Dealer with a Dealer login, you have the option to select whether the orders are to be placed automatically, or you should be prompted to confirm the order before placing. This is explained further below.

To take your strategy to the live market, right-click on the chart, click on Scripts, select the scrip which you want to go live with, and click on Live.



You then need to fill in your order details and set order preferences.

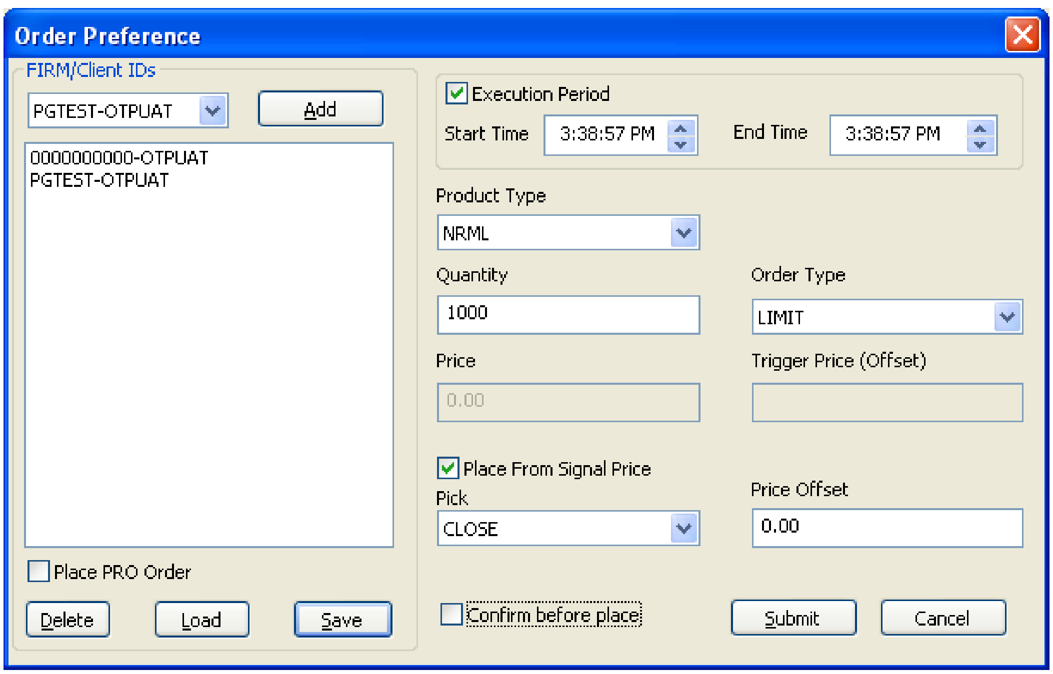


Execution Period can be set to define the Start Time and End Time during which the orders should get placed. Product Type would help to select the product type that has been enabled, similar to placing of normal buy/ sell orders (F1/ F2).

You need to define the quantity to be placed per opportunity that is whenever a condition is met. Order Type would need to be set as Limit/ Market as per preference. In case of selecting a Limit order, you can decide a pre-defined limit or let the limit price be decided based on the signal price, which is when the buy/ sell signal was given.

For placing limit orders based on signal price, tick on the check-box – Place from Signal Price. Select whether the price should be picked as the 'Close', 'Open', 'High' or 'Low' for the interval during which signal was given by selecting the option from the 'Pick' drop-down option. You can also provide an offset that needs to be added to or subtracted from the signal price.

In case of a dealer login, additional features will be available, wherein you can decide whether the orders should get placed automatically or should require confirmation before getting sent to the exchange. As mentioned earlier, for an investor client, the order confirmation will always be asked before it gets placed. The dealer can tick on the check-box for 'Confirm before Place' to get an order prompt on every instance when a condition is met and untick the checkbox 'Confirm before Place' to allow orders to get placed automatically.

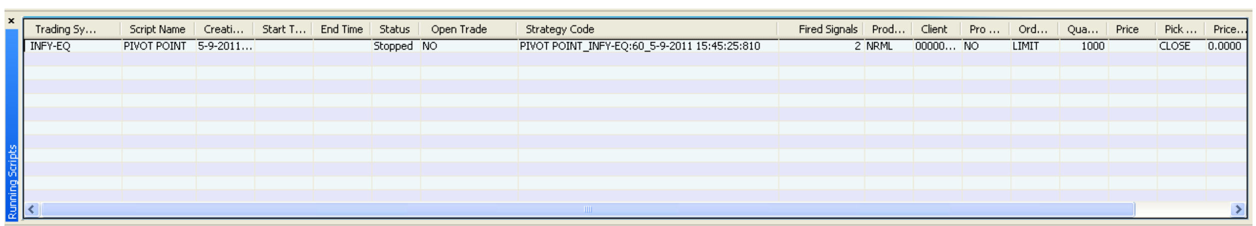


The dealer can also place orders for the strategy, with the same parameters, for multiple accounts mapped to him. In case of trading for multiple accounts, you can save an account list, by clicking on 'Save' and Load this list anytime while going live with any strategy.

Once you click on 'Submit', orders will start getting placed, either automatically or after confirmation depending on user type and order settings, whenever the condition specified in the Script is met with.

Controlling a Live Script:

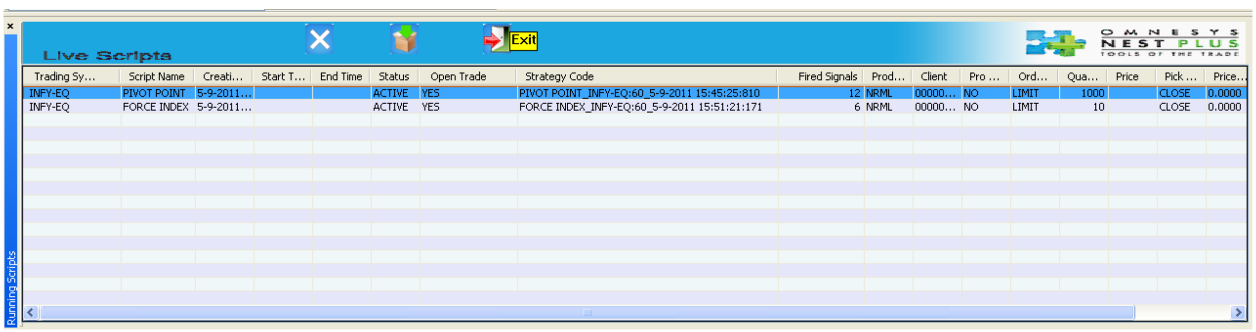
You can view details for all scripts that you have gone live with, by clicking on Pulse Running Scripts under the Nest Plus menu. Alternatively, you can right-click on the chart and select 'Show Running Scripts'. All the scripts that are in 'Live' mode will appear as illustrated below:



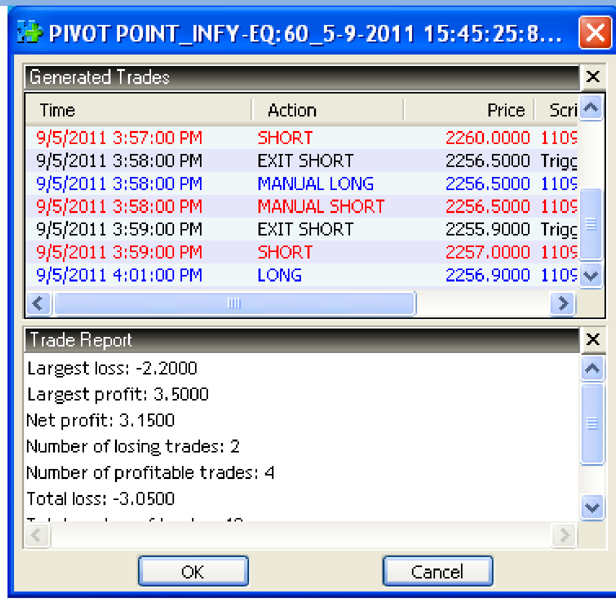
From this window, you can pause/ resume a script, place manual buy/ sell/ square off orders, view reports for all orders placed along with total number of trades, largest profit/ loss, net profit/ loss and so on.

To stop a script, right-click on the 'Running Scripts' window, and click on 'Maximized Mode' to exit maximized mode. You will be able to view an option to stop the strategy. Click on it to pause an active script, and to resume a paused script.

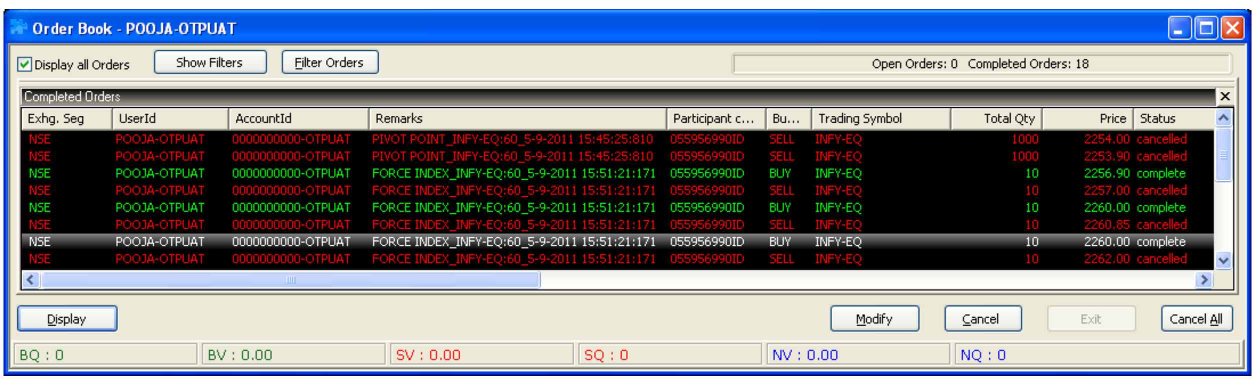
You can also place orders manually overriding the script, by clicking on the Buy or Sell options provided next to the Stop option. In case, you click on Buy, it will get replaced by the Exit option. Similarly, if you click on Sell, it will get replaced by the Exit option to exit any positions taken manually.



To view details of action taken, right-click on the Running Scripts window and click on 'Open Report'.



All orders placed in this manner will be visible in the Order Book (F3 window).



You can distinguish orders placed through Nest Pulse from other orders from the 'Remarks' column in the Order Book. All Order Book features such as modification, cancellation, order history and so on are applicable to orders placed through Nest Pulse as well.

Programming the Language

This guide contains short examples that demonstrate how to perform common, basic tasks such as identifying securities within a specific price range, increase in volatility, crossing over of an indicator, and so forth. You can cut and paste many of these examples right into the **NestPulse**™ programming area in your software.

Also this guide contains a reference of functions, properties, and constants supported by the **NestPulse**™ language as well as hands-on trading system examples. This method of organization allows the beginning programmer to see results immediately while learning at his or her own pace.

NestPulse™ is the engine that drives the scripting language in your trading software. It is a non-procedural scientific vector programming language that was designed specifically for developing trading systems. A *script* is simply a set of instructions that tells the **NestPulse**™ engine to do something useful, such as provide an alert when the price of one stock reaches a new high, crosses over a moving average, or drops by a certain percentage. There are many uses.

Introduction: Important Concepts

NestPulse™ is a powerful and versatile programming language for traders.

The language provides the framework required to build sophisticated trading programs piece by piece without extensive training or programming experience.

The following script is a very simple example that identifies markets that are trading higher than the opening price:

LAST > OPEN

It almost goes without saying that the purpose of this script is to identify when the last price is trading higher than the open price… it is nearly as plain as English.

Just as a spoken language gives you many ways to express each idea, the **NestPulse**™ programming language provides a wide variety of ways to program a trading system. Scripts can be very simple as just shown or extremely complex, consisting of many hundreds of lines of instructions. But for most systems, scripts usually consist of just a few lines of code.

The examples outlined in the first section of this guide are relatively short and simple but provide a foundation for the development of more complex scripts.

Boolean Logic

The scripts shown in this first section may be linked together using Boolean logic just by adding the AND or the OR keyword, for example…

Script 1 evaluates to true when the last price is higher than the open price:

LAST > OPEN

Script 2 evaluates to true when volume is two times the previous day's volume:

VOLUME > REF(VOLUME, 1) * 2

You can aggregate scripts so that your script returns results for securities that are higher than the open *and* with the volume two times the previous volume:

LAST > OPEN AND VOLUME > REF(VOLUME, 1) * 2

Likewise, you can change the AND into an OR to find securities that are either trading higher than the open *or* have a volume two times the previous volume:

LAST > OPEN OR VOLUME > REF(VOLUME, 1) * 2

Once again, the instructions are nearly is plain as the English language. The use of Boolean logic with the AND and OR keywords is a very important concept that is used extensively by the **NestPulse™** programming language.

Program Structure

It does not matter if your code is all on a single line or on multiple lines. It is often easier to read a script where the code is broken into multiple lines. The following script will work exactly as the previous example, but is somewhat easier to read:

LAST > OPEN OR

VOLUME > REF(VOLUME, 1) * 2

It is good practice to structure your scripts to make them as intuitive as possible for future reference. In some cases it may be useful to add *comments* to a very complex script. A comment is used to include explanatory remarks in a script.

Whenever the pound sign is placed at the beginning of a line, the script will ignore the words that follow. The words will only serve as a comment or note to make the script more understandable:

 # Evaluates to true when the last

 # price is higher than the open or the

 # volume is 2 X's the previous volume:

LAST > OPEN OR

VOLUME > REF(VOLUME, 1) * 2

The script runs just as it did before with the only difference being that you can more easily understand the design and purpose of the script.

Functions

The **NestPulse**™ language provides many built-in functions that make programming easier. When functions are built into the core of a programming language they are referred to as *primitives*. The TREND function is one example:

TREND(CLOSE, 30) = UP

In this example, the TREND function tells **NestPulse**™ to identify trades where the closing price is in a 30-day uptrend.

The values that are contained inside a function (such as the REF function or the TREND function) are called *arguments*. Here there are two arguments in the TREND function. Argument #1 is the closing price, and argument #2 is 30, as in "30 days" or "30 periods".

Only one of two things will occur if you use a function incorrectly will automatically fix the problem and the script will still run, or **NestPulse**™ will report an error, tell you what's wrong with the script, and then allow you to fix the problem and try again.

In other words, user input errors will never cause **NestPulse**™ to break or return erroneous results without first warning you about a potential problem.

Let's take CLOSE out of the TREND function and then try to run the script again:

TREND(30) = UP

The following error occurs:

Error: argument of 'TREND' function not optional.

We are given the option to fix the script and try again.

Vector Programming

Vector programming languages (also known as *array* or *multidimensional* languages) generalize operations on scalars to apply transparently to vectors, matrices, and higher dimensional arrays.

The fundamental idea behind vector programming is that operations apply at once to an entire set of values (a *vector* or *field*). This allows you to think and operate on whole aggregates of data, without having to resort to explicit loops of individual scalar operations.

As an example, to calculate a simple moving average based on the median price of a stock over 30 days, in a traditional programming language such as BASIC you would be required to write a program similar to this:

For each symbol

For bar = 30 to max

Average = 0

For n = bar – 30 to bar

median = (CLOSE + OPEN) / 2

Average = Average + median

Next

MedianAverages(bar) = Average / 30

Next bar

Next symbol

Nine to ten lines of code would be required to create the "MedianAverages" vector. But with **NestPulse**™, you can effectively accomplish the same thing using only one line:

SET MedianAverage = SimpleMovingAverage((CLOSE + OPEN) / 2, 30)

And now MedianAverage is actually a new vector that contains the 30-period simple moving average of the median price of the stock at each point.

It is not uncommon to find array programming language “one-liners” that require more than a couple of pages of BASIC, Java or C++ code.

The REF Function

At this point you may be wondering what “REF” and “TREND” are. These are two of the very useful primitives that are built into the **NestPulse**™ language.

The REF function is used whenever you want to reference a value at any specific point in a vector. Assume the MedianAverage vector contains the average median price of a stock. In order to access a particular element in the vector using a traditional programming language, you would write:

SET A = MedianAverage[n]

Using **NestPulse**™ you would write:

SET A = REF(MedianAverage, n)

The main difference other than a variation in syntax is that traditional languages reference the points in a vector starting from the beginning, or 0 if the vectors are zero-based. **NestPulse**™ on the other hand references values backwards, from the end. This is most convenient since the purpose of **NestPulse**™ is of course, to develop trading systems. It is always the *last,* most *recent* value that is of most importance. To get the most recent value in the MedianAverage vector we could write:

SET A = REF(MedianAverage, 0)

Which is the same as not using the REF function at all. Therefore the preferred way to get the last value (the most recent value) in a vector is to simply write:

SET A = MedianAverage

The last value of a vector is always assumed when the REF function is absent.

To get the value as of one bar ago, we would write: SET A = REF(MedianAverage, 1)

Or two bars ago:

SET A = REF(MedianAverage, 2)

The TREND Function

Stock traders often refer to "trending" as a state when the price of a stock has been increasing (up-trending) or decreasing (down-trending) for several days, weeks, months, or years. The typical investor or trader would avoid opening a new long position of a stock that has been in a *downtrend* for many months.

NestPulse™ provides a primitive function aptly named TREND especially for detecting trends in stock price, volume, or indicators: TREND(CLOSE, 30) = UP

This tells **NestPulse**™ to identify trades where the closing price is in a 30-day uptrend. Similarly, you could also use the TREND function to find trends in volume or technical indicators:

the volume has been

in a downtrend for at least 10 days:

TREND(VOLUME, 10) = DOWN

the 14-day CMO indicator

has been up-trending for at least 20 days:

TREND(CMO(CLOSE, 14), 20) = UP

It is useful to use the TREND function for confirming a trading system signal.

Suppose we have a trading system that buys when the close price crosses above a 20-day Simple Moving Average. The script may look similar to this:

```
# Gives a buy signal when the close price crosses above the 20-day SMA
CROSSOVER(CLOSE, SimpleMovingAverage(CLOSE, 20)) = TRUE
```

It would be helpful in this case to narrow the script down to only the securities that have been in a general downtrend for some time. We can add the following line of code to achieve this:

```
AND TREND(CLOSE, 40) = DOWN
```

TREND tells us if a vector has been trending upwards, downwards, or sideways, but does not tell us the degree of which it has been trending. We can use the REF function in order to determine the range in which the data has been trending. To find the change from the most current price and the price 40 bars ago, we could write:

```
SET A = LAST – REF(CLOSE, 40)
```

Price Gaps and Volatility

Although the TREND function can be used for identifying trends and the REF function can be used for determining the degree in which a stock has moved, it is often very useful to identify gaps in prices and extreme volume changes, which may be early indications of a change in trend.

We can achieve this by writing:

Returns true when the price has gapped up

LOW > REF(HIGH, 1)

Or:

Returns true when the price has gapped down

HIGH < REF(LOW, 1)

You can further specify a minimum percentage for the price gap:

Returns true when the price has gapped up at least 1%

LOW > REF(HIGH, 1) * 1.01

And with a slight variation we can also the volume is either up or down by a large margin:

the volume is up 1000%

VOLUME > REF(VOLUME, 1) * 10

Or by the average volume:

the volume is up 1000% over average volume

VOLUME > SimpleMovingAverage(VOLUME, 30) * 10

We can also measure volatility in price or volume by using any one of the built-in technical indicators such as the Volume Oscillator, Chaikin Volatility Index, Coefficient of Determination, Price Rate of Change, Historical Volatility Index, etc. These technical indicators are described in the following chapters.

TECHNICAL ANALYSIS

NestPulse™ provides many built-in technical analysis functions. Using only a single line of code you can calculate functions such as Moving Averages, Bollinger Bands, Japanese Candlesticks, and so on. A complete list of technical analysis functions is covered in chapter 3.

The following is a simple example of how to use one of the most common technical analysis functions, the simple moving average:

LAST > SimpleMovingAverage(CLOSE, 20)

The script will check if the last price is over the 20-day moving average of the close price.

The CLOSE variable is actually a vector of closing prices, not just the most recent close price. You can use the OPEN, HIGH, LOW, CLOSE and VOLUME vectors to create your own calculated vectors using the SET keyword:

SET Median = (CLOSE + OPEN) / 2

This code creates a vector containing the median price for each trading day.

We can use the Median vector inside any function that requires a vector:

LAST > SimpleMovingAverage(Median, 20)

And this evaluates to true when the last price is greater than a 20-day moving average of the median price.

Because functions return vectors, functions can also be used as valid arguments within other functions:

LAST > SimpleMovingAverage(SimpleMovingAverage(CLOSE, 30), 20)

This evaluates to true when the last price is greater than the 20-day moving average of the 30-day moving average of the close price.

Crossovers

You may be familiar with the term “crossover”, which is what happens when one series crosses over the top of another series as depicted in the image on the right.

Many technical indicators such as the MACD for example, have a “signal line”. A buy or sell signal is generated when the signal line crosses over or under the technical indicator.

The CROSSOVER function helps you one series has crossed over another.

For example, we can find the exact point in time when one moving average crossed over another by using the CROSSOVER function:

SET MA1 = SimpleMovingAverage(CLOSE, 28)

SET MA2 = SimpleMovingAverage(CLOSE, 14)

CROSSOVER(MA1, MA2) = TRUE

The script above will evaluate to true when the MA1 vector most recently crossed over the MA2 vector. And we can reverse the script to the MA1 vector crossed *below* the MA2 vector:

CROSSOVER(MA2, MA1) = TRUE

Primitive Functions & Operators

Primitives

The built-in functions of **NestPulse**™, also known as *primitives*. These important functions define the **NestPulse**™ programming language and provide the basic framework required to build complex trading systems from the ground up.

Literally any type of trading system can be developed using the **NestPulse**™ programming language with minimal effort. If a system can be expressed in mathematical terms or programmed in any structured, procedural language such as C++, VB, or Java for example, you can rest assured that the same formulas can also be programmed using the **NestPulse**™ programming language.

Sometimes technical analysis formulas can be very complex. For example, technical analysis functions exist that require recursive calculations and complicated IF-THEN-ELSE structures as part of their formula. These complex trading systems are traditionally developed in a low level programming language.

This chapter outlines how **NestPulse**™ can be used to perform these same calculations in a much simpler way by means of vector operations and simulated control structure.

Conditional "IF" Function

IF(Condition, True part, False part)

The conditional "IF" function allows you to design complex Boolean logic filters. If you paste the following script into the Script area in your trading software application, you will see a column of numbers that oscillate between 1 and -1, depending on when the closing price is greater than the opening price:

SET A = IF(CLOSE > OPEN, 1, -1)

The first argument of the “IF” function is a logical test. The second argument is the value that will be used if the condition evaluates to TRUE. Conversely, the third argument is the value that will be used if the condition evaluates to FALSE.

The logical test may be any value or expression that can be evaluated to TRUE or FALSE. For example, CLOSE = OPEN is a logical expression; if the close price is the same as the opening price, the expression evaluates to TRUE.

Otherwise, the expression evaluates to FALSE.

LOOP Function

LOOP(Vector1, Vector2, Offset1, Offset2, Operator)

LOOP provides simulated control structure by means of a single function call.

Consider the following:

SET X = CLOSE

SET X = REF(X, 1) + X

This script simply adds the previous close to the most current close. REF(X, 1) is evaluated once. This is expected behavior for a vector programming language; vectors are calculated independently in a stepwise fashion and are not recursive.

Now by changing CLOSE to 0, logically we would expect X to equal the previous X value plus one, and therefore expect REF(X, 1) to be evaluated once for each record in the vector:

SET X = 0

SET X = REF(X, 1) + X

Although we are looking at the exact same formula, because we are initializing X with a scalar and X is not related to any existing vector we would now expect X to be calculated as a series: 1,2,3,4,5,6,...n

We are now exceeding the limits of a vector programming language by requiring control structure.

Anytime we assign a variable to itself such as SET X = F(X) we are expecting F(X) to be recursive.

In the first example we write SET X = CLOSE. CLOSE is a variable, not a function and does not have any relationship with X. Our expectations change when we initialize X with anything other than an existing vector.

The LOOP function overcomes this inherent limitation by simulating a structured programming construct, the for-loop iteration:

LOOP(Vector1, Vector2, Offset1, Offset2, Operator)

Vector1 is the vector to initialize the calculation from. Offset1 is the offset where values are referenced in Vector1 for the incremental calculation, and Offset2 is the offset where values are referenced from in Vector2.

Example 1:

X (Vector1) is a series from 5.25 to 11.25.

If we write LOOP(X, 2, 1, 0, MULTIPLY) the vector returned will contain values initialized by X, offset by 1 and multiplied by 2:

Example 2:

In the case of SET X = REF(X, 1), Vector1 is X and Vector2 is 1. Since we're adding the value of 1 (not a vector) to X in the following example, Offset2 is set to zero:

SET X = LOOP(X, 1, 1, 0, ADD)

And now X contains the series 1,2,3,4,5,6,...n

Example 3:

SET X = REF(CLOSE,1)

SET Y = (REF(Y, 3) – X) * 2

Because Y requires control structure we must instead write: SET X = REF(CLOSE,1)

SET Y = LOOP(Y, X, 3, 0, SUBTRACT) * 2

We could reduce that to:

SET Y = LOOP(Y, CLOSE, 3, 1, SUBTRACT) * 2

Valid operators are ADD, SUBTRACT, MULTIPLY and DIVIDE .

X	LOOP
5.25	5.25
6.25	10.5
7.25	21
8.25	42
9.25	84
10.25	168
11.25	336

COUNTIF

COUNTIF(Condition)

Returns a vector representing the total number of times the specified condition evaluated to TRUE.

Example:

COUNTIF(CROSSOVER(SimpleMovingAverage(CLOSE, 14), CLOSE)) The script returns a vector with increasing values expressing the number of times the 14-day Simple Moving Average crossed over the closing price.

LASTIF

LASTIF(Condtion)

Similar to COUNTIF, except LASTIF returns a vector containing the number of days since the last time the specified condition evaluated to TRUE. The count is reset to zero each time the condition evaluates to TRUE.

Example:

LASTIF(CLOSE < REF(CLOSE, 1))

The script returns a vector that increases in value for each bar where the closing price was *not* less than the previous closing price. When the condition evaluates to TRUE, meaning the closing price was less than the previous closing price, the reference count is reset to zero.

SUMIF

SUMIF(Condtion, Vector)

Last in the “IF” function lineup is the SUMIF function. This function outputs a running sum of all values in the supplied Vector wherever the supplied Condition evaluates to TRUE.

For example if we wanted a vector containing the sum of volume for all the days where the closing price closed up 5%, we could write:

SUMIF(CLOSE > REF(CLOSE,1) * 1.05, VOLUME)

The result will be a vector containing a running sum of volume for each day where the closing price closed up at least 5%.

SUM

SUM(Vector, Periods)

The SUM function (not to be confused with the SUMIF function) outputs a vector containing a running sum, as specified by the Periods argument.

Example:

SUM(CLOSE, 10)

The script returns a vector of sums based on a 10-period window.

AVG

AVERAGE(Vector, Periods)

AVG(Vector, Periods)

Returns a vector containing a running average, as specified by the Periods argument. The AVERAGE function can also be referenced by AVG for short.

Example:

AVERAGE(CLOSE, 10)

AVG(CLOSE, 10)

Both scripts return a vector of averages based on a 10- period window.

MAX

MAX(Vector, Periods)

Returns a vector containing a running maximum, as specified by the Periods argument. The values represent the maximum value for each window.

Example:

MAX(CLOSE, 10)

Returns a vector of maximum values based on a 10- period window.

MIN

MIN(Vector, Periods)

Returns a vector containing a running minimum, as specified by the Periods argument. The values represent the minimum value for each window.

Example:

MIN(CLOSE, 10)

Returns a vector of minimum values based on a 10- period window.

MAXOF

MAXOF(Vector1, Vector2, [Vector3]…[Vector8])

Returns a vector containing a maximum value of all specified vectors, for up to eight vectors. Vector1 and Vector2 are required and vectors 3 through 8 are optional.

Example:

MAXOF(CLOSE, OPEN)

Returns a vector containing the maximum value for each bar, which is either the opening price or the closing price in this example.

MINOF

MINOF(Vector1, Vector2, [Vector3]…[Vector8])

Returns a vector containing a minimum value of all specified vectors, for up to eight vectors. Vector1 and Vector2 are required and vectors 3 through 8 are optional.

Example:

MINOF(CLOSE, OPEN)

Returns a vector containing the minimum value for each bar, which is either the opening price or the closing price in this example.

REF

REF(Vector, Periods)

By default all calculations are performed on the last, most recent value of a vector. The following script evaluates to true when the last open price (the *current* bar's open price) is less than $30:

OPEN < 30

OPEN is assumed to be the *current* bar's open by default. You can reference a previous value of a vector by using the REF function: REF(OPEN, 1) < 30

And now the script will *previous* bar's open price was less than $30. The number 1 (the second argument) tells the REF function to reference values as of one bar ago. To reference values two bars ago, simply use 2 instead of 1. The valid range for the Periods argument is 1 – 250 unless otherwise noted.

TREND

TREND(Vector)

The TREND function can be used to determine if data is trending upwards, downwards, or sideways. This function can be used on the price (open, high, low, close), volume, or any other vector. The TREND function returns a constant of either UP, DOWN or SIDEWAYS. Example:

TREND(CLOSE) = UP AND TREND(VOLUME) = DOWN

TREND is often the first function used as a means of filtering securities that are not trending in the desired direction.

CROSSOVER

Many technical indicators such as the MACD for example, have a “signal line”.

Traditionally a buy or sell signal is generated when the signal line crosses over or under the technical indicator.

The CROSSOVER function helps you one series has crossed over another. For example, we can find the exact point in time when one moving average crossed over another by using the CROSSOVER function: SET MA1 = SimpleMovingAverage(CLOSE, 28)

SET MA2 = SimpleMovingAverage(CLOSE, 14)

CROSSOVER(MA1, MA2) = TRUE

The script above will evaluate to true when the MA1 vector most recently crossed over the MA2 vector. And we can reverse the script to the MA1 vector crossed *below* the MA2 vector:

CROSSOVER(MA2, MA1) = TRUE

Math Functions

Note that all math functions return a vector. For example ABS(CLOSE – OPEN) returns a vector of the ABS value of CLOSE – OPEN (one record per bar). The RND function returns a vector of random values, one for each bar, and so forth.

ABS

The ABS function returns the absolute value for a number. Negative numbers become positive and positive numbers remain positive.

Example:

ABS(CLOSE – OPEN)

The script always evaluates to a positive number, even if the opening price is greater than the closing price.

SIN

The SIN function returns the sine for a number (angle).

Example:

SIN(45)

The script outputs 0.851

COS

COS returns the cosine for a number (angle).

Example:

COS(45)

The script outputs 0.525

TAN

The TAN function returns the tangent for a number (angle).

Example:

TAN(45)

The script outputs 1.619

ATN

Returns the arctangent for a number.

Example:

ATN(45)

The script outputs 1.548

EXP

EXP raises *e* to the power of a number. The LOG function is the reverse of this function.

Example:

EXP(3.26)

The script outputs 26.28

LOG

Returns the natural logarithm of a positive number. The EXP function is the reverse of this function. Also see LOG10.

Example:

LOG(26.28)

The script outputs 3.26

LOG10

Returns the base 10 logarithm of a positive number. Also see LOG.

Example:

LOG10(26.28)

The script outputs 1.42

RND

The RND function returns a random number from 0 to a maximum value.

Example:

RND(100)

Outputs a random number from 0 to 100.

Operators

Equal (=)

The equal operator is used to assign a value to a variable or vector, or to compare values. When used for assignment, a single variable or vector on the left side of the = operator is given the value determined by one or more variables, vectors, and/or expressions on the right side. Also, the SET keyword must precede the variable name when the = operator is used for an assignment: SET A = 123

SET B = 123

A = B = TRUE

Greater Than (>)

The > operator determines if the first expression is greater-than the second expression.

Example:

SET A = 124

SET B = 123

A > B = TRUE

Less Than (<)

The < operator determines if the first expression is less-than the second expression.

Example:

SET A = 123

SET B = 124

A > B = TRUE

Greater Than Or Equal To (>=)

The >= operator determines if the first expression is greater-than or equal to the second expression.

Example:

SET A = 123

SET B = 123

A >= B = TRUE

And:

SET A = 124

SET B = 123

A >= B = TRUE

Less Than Or Equal To (<=)

The <= operator determines if the first expression is less-than or equal to the second expression.

Example:

SET A = 123

SET B = 123

A <= B = TRUE

And:

SET A = 123

SET B = 124

A <= B = TRUE

Not Equal (<> or !=)

Both the != and the <> inequality operators determine if the first expression is not equal to the second expression.

Example:

SET A = 123

SET B = 124

A != B = TRUE

AND

The AND operator is used to perform a logical conjunction on two expressions, where the expressions are Null, or are of Boolean subtype and have a value of True or False.

The AND operator can also be used a “bitwise operator” to make a bit-by-bit comparison of two integers. If both bits in the comparison are 1, then a 1 is returned. Otherwise, a 0 is returned.

When using the AND to compare Boolean expressions, the order of the expressions is not important.

Example:

(TRUE = TRUE AND FALSE = FALSE) = TRUE

And:

(TRUE = TRUE AND FALSE = TRUE) = FALSE

OR

The OR operator is used to perform a logical disjunction on two expressions, where the expressions are Null, or are of Boolean subtype and have a value of True or False.

The OR operator can also be used a “bitwise operator” to make a bit-by-bit comparison of two integers. If one or both bits in the comparison are 1, then a 1 is returned. Otherwise, a 0 is returned.

When using the OR to compare Boolean expressions, the order of the expressions is important.

Example:

(TRUE = TRUE OR TRUE = FALSE) = TRUE

And:

(FALSE = TRUE OR TRUE = FALSE) = FALSE

XOR

The XOR operator is used to perform a logical exclusion on two expressions, where the expressions are Null, or are of Boolean subtype and have a value of True or False.

The XOR operator can also be used a “bitwise operator” to make a bit-by-bit comparison of two integers. If both bits are the same in the comparison (both are 0’s or 1’s), then a 0 is returned. Otherwise, a 1 is returned.

Example:

(TRUE XOR FALSE) = TRUE

And:

(FALSE XOR FALSE) = FALSE

NOT

The NOT operator is used to perform a logical negation on an expression. The expression must be of Boolean subtype and have a value of True or False. This operator causes a True expression to become False, and a False expression to become True.

Example:

NOT (TRUE = FALSE) = TRUE

And:

NOT (TRUE = TRUE) = FALSE

EQV

The EQV operator is used to perform a logical comparison on two expressions (I.e., are the two expressions identical), where the expressions are Null, or are of Boolean subtype and have a value of True or False.

The EQV operator can also be used a “bitwise operator” to make a bit-by-bit comparison of two integers. If both bits in the comparison are the same (both are 0’s or 1’s), then a 1 is returned. Otherwise, a 0 is returned.

The order of the expressions in the comparison is not important.

Example:

TRUE EQV TRUE = TRUE

And:

TRUE EQV FALSE = FALSE

MOD

The MOD operator divides two numbers and returns the remainder. In the example below, 5 divides into 21, 4 times with a remainder of 1.

Example:

21 MOD 5 = 1

And:

Trading Systems

A trading system is basically a set of rules that determine entry and exit points for any given stock. Traders often refer to these points as *trade signals*.

A trading system is objective and mechanical. The purpose is to provide a strategy to produce profits greater than losses by controlling your trades for you.

This chapter provides hands-on learning by teaching the trader how to translate trading system rules into script form using real trading systems as examples.

Trading systems usually include one or more technical indicators in their implementation. For example, a *Moving Average Crossover* system would buy when a short-term moving average crosses above a long-term moving average and sell when a short-term moving average crosses below a long-term moving average.

Trading systems may have any number of rules, such as “don’t buy unless volume is trending upwards”, or “exit if Parabolic SAR crosses the close”, etc.

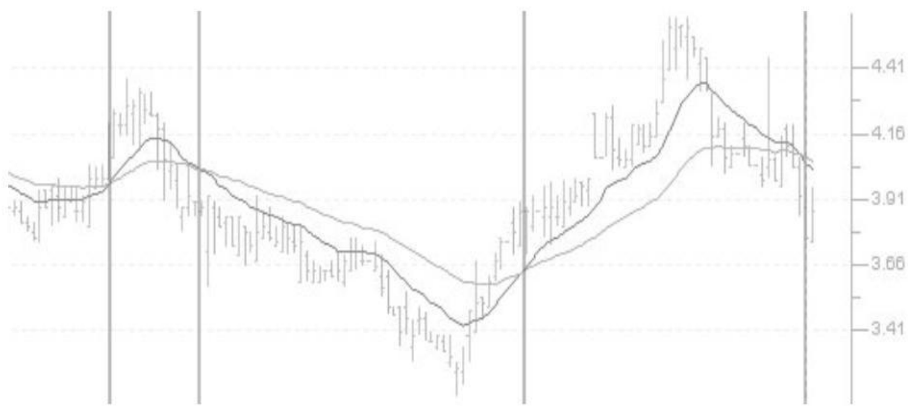
The actual profitability of a trading system depends on how well the trading system’s rules perform on a trade-by-trade basis. Traders spend much of their time optimizing their trading systems in order to increase profits and reduce risks.

In the case of a basic *Moving Average Crossover* system, this is accomplished by modifying the parameters of the moving averages themselves.

A trader may optimize a trading system by means of *back testing*. The back testing feature of **NestPulse**™ allows you to back test your trading systems and modify parameters to achieve the maximum amount of profit and minimum amount of risk. Refer to your trading software documentation for details.

Moving Average Crossover System

The Moving Average Crossover System is perhaps the simplest of all trading systems. This system uses two moving averages to generate signals. A buy signal is generated when a short-term moving average crosses over a longerterm moving average, and sells when a short-term moving average crosses below a long-term moving average.



The number of signals generated by this trading system is proportional to the length *and type* of moving averages used. Short-term moving averages generate more signals and enter into trades sooner than longer-term moving averages.

Unfortunately, a very short-term moving average crossover system will also generate more false signals than a longer-term system, while a very long-term system will generate fewer false signals, but will also miss a larger proportion of profits. This difficult balance applies to nearly every trading system and is the core subject of numerous books on technical analysis.

One solution to this problem is to use a secondary technical indicator to confirm entry and/or exit signals. A popular technical indicator used primarily for exit signals is the Parabolic SAR. The following script uses a 20/60 EMA for entries and a Parabolic SAR for exits.

Moving Average Crossover System Script

Buy Signals

20-period EMA crosses over the 60-period EMA

CROSSOVER(EMA(CLOSE, 20), EMA(CLOSE, 60))

Sell Signals

20-period EMA crosses under the 60-period EMA

CROSSOVER(EMA(CLOSE, 60), EMA(CLOSE, 20))

Exit Long

The close crosses above the Parabolic SAR

CROSSOVER(CLOSE, PSAR(CLOSE, 0.02, 0.2))

Exit Short

The close crosses below the Parabolic SAR

CROSSOVER(PSAR(CLOSE, 0.02, 0.2), CLOSE)

Sample Script 1 (For Bullish Markets)

Buy Signals

10-period EMA crosses over the 30-period EMA and a positive MACD

(EMA(CLOSE,10) > EMA(CLOSE,30)) AND (MACD(6,12,9,EXPONENTIAL) > 0)

Exit Long

30-period EMA crosses over the 10-period EMA and a negative MACD

(EMA(CLOSE,10) < EMA(CLOSE,30)) AND (MACD(6,12,9,EXPONENTIAL) < 0)

Sample Script 2 (For Bullish Markets)

Buy Signals

A combination of Simple moving average,MACD and stochastic.

(SMA(CLOSE,2) > SMA(CLOSE,10)) AND MACD(6,12,9,SIMPLE) > 0 AND (SOPK(9, 3, 9, SIMPLE) > 80 OR SOPD(9, 3, 9, SIMPLE) > 80)

Exit Long

A combination of Simple moving average,MACD and stochastic

(SMA(CLOSE,2) < SMA(CLOSE,10)) AND MACD(6,12,9,SIMPLE) < 0 AND (SOPK(9, 3, 9, SIMPLE) < 20 OR SOPD(9, 3, 9, SIMPLE) < 20)

Price Gap System

An upward price gap occurs when a stock opens substantially higher than the previous day's high price. This often occurs after an unexpected announcement, much better than expected earnings report, and so forth.

A large number of buy orders are executed when the market opens. During this time the price may be exaggerated as investors may be buying the stock simply because it shows strength at the opening.

The price often retreats to fill the gap once orders stop coming in and the demand for the stock subsides. The key to this trading system is that reversals usually occur during the *first hour* of trading. In other words, if the gap is not filled during the first hour then we may assume that buying will continue.

This trading system is often more successful if volume is around twice the fiveday average of volume.

Example: The script returns securities that have gapped up by 2% and closed near the high. When the market opens on the following day, the strategy would be to buy stock after the first hour of trading if the strength sustained.

A stop-loss order would be set at the day's low. A conservative profit objective would normally be half the percentage of the gap, or 1% in this case.

Price Gap Script

Buy Signals

A 2% gap up in price over the previous day on high volume

LOW > REF(HIGH,1) * 1.02 AND VOLUME > SMA(VOLUME, 5) * 2

Sell Signals

A 2% gap down in price over the previous day on high volume

HIGH < REF(LOW,1) * 0.98 AND VOLUME > SMA(VOLUME, 5) * 2

Exit Long

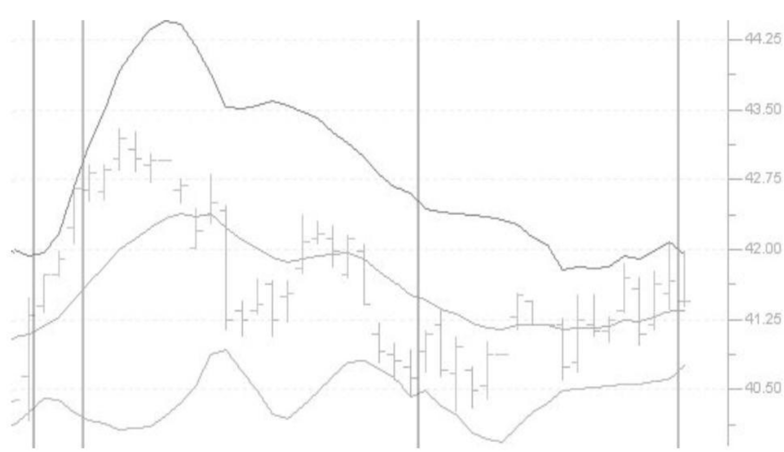
Use a profit objective roughly ½ the size of the gap with a stop-loss.

Exit Short

Use a profit objective roughly ½ the size of the gap with a stop-loss.

Bollinger Bands System

Bollinger bands are similar to moving averages except they are shifted above and below the price by a certain number of standard deviations to form an envelope around the price. And



unlike a moving average or a moving average envelope, Bollinger bands are calculated in such a way that allows them to widen and contract based on market volatility.

Prices usually stay contained within the bands. One strategy is to buy or sell after the price touches and then retreats from one of the bands. A move that originates at one band usually tends to move all the way to the other band.

Another strategy is to buy or sell if the price goes outside the bands. If this occurs, the market is likely to continue in that direction for some length of time.

The Bollinger band trading system outlined in this example uses a combination of both trading strategies. The system buys if a recent bar touched the bottom band and the current bar is within the bands, and also buys if the current high has exceeded the top band by a certain percentage. The system sells based on the opposite form of this strategy.

Bollinger Bands Script

Buy Signals

```
# Buy if a previous value was below the low band and is now above
SET Bottom = BBB(CLOSE, 20, 2, EXPONENTIAL)
SET Top = BBT(CLOSE, 20, 2, EXPONENTIAL)
((REF(CLOSE, 1) < REF(Bottom, 1)) AND
CLOSE > Bottom) OR
# Also buy if the close is above the top band plus 2%
CLOSE > Top * 1.02
```

Sell Signals

```
# Sell if a previous value was above the high band and is now below
SET Bottom = BBB(CLOSE, 20, 2, EXPONENTIAL)
SET Top = BBT(CLOSE, 20, 2, EXPONENTIAL)
```

((REF(CLOSE, 1) > REF(Top, 1)) AND

CLOSE < Top) OR

Also sell if the close is below the bottom band minus 2%

CLOSE < Bottom * 0.98

Historical Volatility and Trend

This trading system buys or sells on increasing volume and lessening volatility.

The concept is that trends are more stable if volatility has been decreasing and volume has been increasing over many days.

Volume is an important component to this trading system since almost every important turning point in a stock is accompanied by an increase in volume.

The key element in this trading system is the length of the primary price trend.

The longer the price trend is, the more reliable the signal.

Also try experimenting with this trading system by substituting the TREND function for volume with the Volume Oscillator function, or the Volume Rate of Change function.

Historical Volatility and Trend Script

Buy Signals

Buy if volatility is decreasing and volume is increasing with price in an uptrend

HistoricalVolatility(CLOSE, 15, 252, 2) < REF(HistoricalVolatility(CLOSE, 15, 365, 2), 15) AND

TREND(VOLUME, 5) = UP AND TREND(CLOSE, 40) = UP

Sell if volatility is decreasing and volume is increasing with price in a downtrend

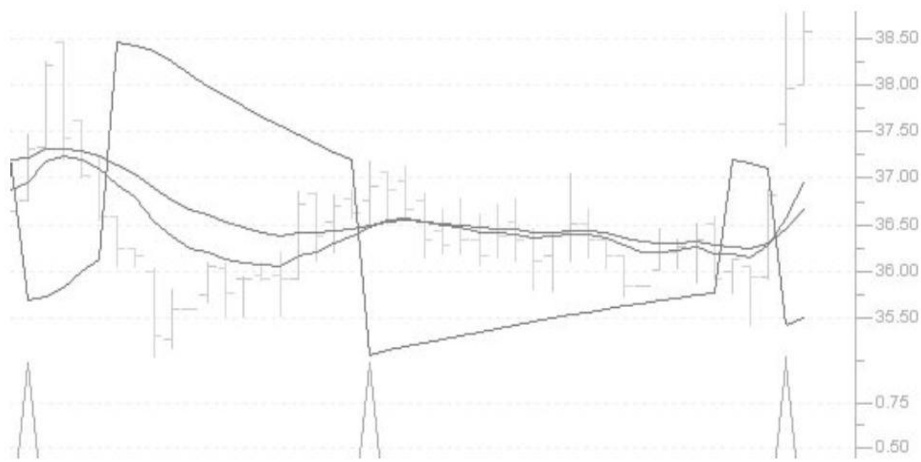
HistoricalVolatility(CLOSE, 15, 252, 2) < REF(HistoricalVolatility(CLOSE, 15, 365, 2), 15) AND

TREND(VOLUME, 5) = UP AND TREND(CLOSE, 40) = DOWN

Parabolic SAR / MA System

This system is a variation of a standard moving average crossover system.

Normally a Parabolic SAR is used only as a signal for exit points, however in this trading system we use the crossover of two exponential moving averages to decide if we should buy or sell whenever the Parabolic SAR indicator crosses over the close.



The Parabolic SAR can be used in the normal way after the trade has been opened. Profits should be taken when the close crosses the Parabolic SAR.

This example shows how to use Boolean logic to find securities that match the condition either for the current trading session or the previous trading day.

Parabolic SAR / MA Script

Buy Signals

Buy if the Mas crossed today or yesterday and

if the PSAR crossed today or yesterday

```
FIND STOCKS WHERE
(CROSSOVER(CLOSE, PSAR(0.02, 0.2)) OR
CROSSOVER(REF(CLOSE,1), PSAR(0.02, 0.2))) AND
(CROSSOVER(EMA(CLOSE, 10), EMA(CLOSE, 20)) OR
CROSSOVER(REF(EMA(CLOSE, 10),1), REF(EMA(CLOSE, 20),1)))
```

Sell Signals

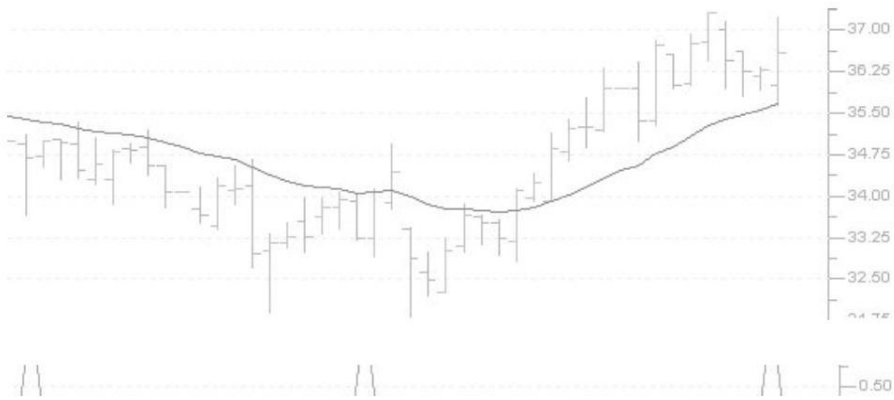
Sell if the Mas crossed today or yesterday and

if the PSAR crossed today or yesterday

```
FIND STOCKS WHERE
(CROSSOVER(PSAR(0.02, 0.2), CLOSE) OR
CROSSOVER(PSAR(0.02, 0.2), REF(CLOSE,1))) AND
(CROSSOVER(EMA(CLOSE, 20), EMA(CLOSE, 10)) OR
CROSSOVER(REF(EMA(CLOSE, 20),1), REF(EMA(CLOSE, 10),1)))
```

MACD Momentum System

In this trading system we use an exponential moving average and the TREND function to identify market inertia, and we use the Moving Average Convergence / Divergence (MACD) indicator to detect market momentum.



As you may know, the MACD indicator reflects the change of power between traders who are on the long side and traders who are on the short side. When the trend of the MACD indicator goes up, it indicates that the market is predominated by bulls, and when it falls, it indicates that bears have more influence. This is known as market momentum.

This system buys when both inertia (a TREND of the EMA) and momentum (the MACD) are both in favor of rising prices. The system sells when the reverse is true.

Exit signals are generated whenever either signal turns to the opposite direction.

MACD Momentum Script

Buy Signals

Buy if both momentum and inertia are favorable

TREND(EMA(CLOSE, 20), 15) = UP AND

TREND(MACD(13, 26, 9, SIMPLE), 5) = UP

Sell Signals

Sell if both momentum and inertia are favorable

TREND(EMA(CLOSE, 20), 15) = DOWN AND

TREND(MACD(13, 26, 9, SIMPLE), 5) = DOWN

Exit Long Signal

Exit if either momentum or inertia become unfavorable

TREND(EMA(CLOSE, 20), 15) = DOWN OR

TREND(MACD(13, 26, 9, SIMPLE), 5) = DOWN

Exit Short Signal

Exit if either momentum or inertia become unfavorable

TREND(EMA(CLOSE, 20), 15) = UP OR

TREND(MACD(13, 26, 9, SIMPLE), 5) = UP

Narrow Trading Range Breakout

Stocks that remain bound by narrow trading ranges often tend to continue in the direction of their breakout. That is to say, if a stock remains in a narrow range between $40 and $45 for an extended period then breaks above $50, it is likely to continue rising for the foreseeable future. The premise being that the longer a stock remains in a tight range, the more difficult it is becomes to breakout of the trading range. Therefore when the breakout occurs, the uptrend should continue.

Narrow Trading Range Script

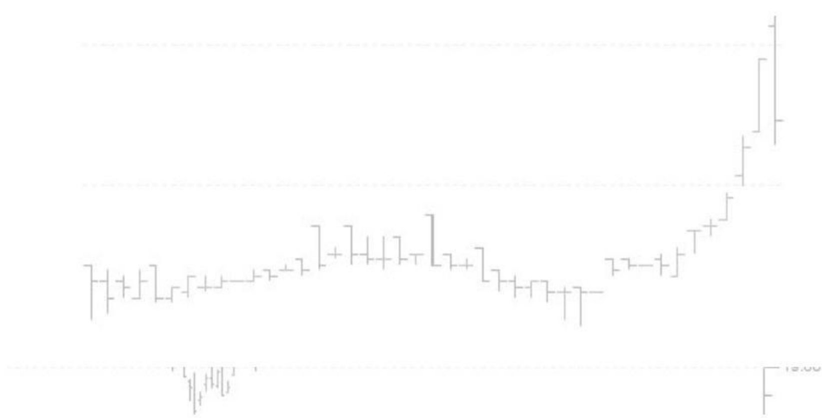
```
# Define a 2% trading range over 50 days
FIND STOCKS WHERE
MAX(CLOSE, 50) < CLOSE * 1.01 AND
MIN(CLOSE, 50) > CLOSE * 0.98 AND
```

Filter out inactive securities

```
CLOSE != REF(CLOSE, 1) AND
REF(CLOSE,1) != REF(CLOSE, 2) AND
REF(CLOSE,2) != REF(CLOSE, 3)
```

Outside Day System

An Outside Day occurs when the current bar's high price is higher than the previous bar's high price, and the current bar's low price is lower than the previous bar's low price. The close must be opposite of the trend (if the trend is up, the close must be lower than the open). Outside days occur frequently and may be used as part of a short term trading strategy.



Outside days that occur after a strong uptrend as shown in this chart indicate market indecision, and may signal a reversal or temporary correction in price.

Depending on market direction, outside days can be either bullish or extremely bearish. If the reversal occurs at the stock's resistance level, it is interpreted as bearish. If it occurs at the stock's support level, it is interpreted as bullish.

Outside Day Script

Buy Signals

```
# Find outside days
LOW < REF(LOW, 1) AND
HIGH > REF(HIGH, 1) AND
HIGH > REF(HIGH, 1) AND
CLOSE < OPEN AND
# Outside days are more significant if the
# previous bar is shorter in height
HIGH – LOW > (REF(HIGH, 1) – REF(LOW, 1)) * 1.5 AND
# The trend should be up
TREND(CLOSE, 30) = UP
```

Sell Signals

```
# Find outside days
LOW < REF(LOW, 1) AND
HIGH > REF(HIGH, 1) AND
HIGH > REF(HIGH, 1) AND
CLOSE < OPEN AND
HIGH – LOW > (REF(HIGH, 1) – REF(LOW, 1)) * 1.5 AND
```

The trend should be down for a sell signal

TREND(CLOSE, 30) = DOWN

	Indicator	Signature	Description	Sample script	Remarks
1	Simple Moving Average	SMA(Vector, Periods)	The Simple Moving Average is simply an average of values over a specified period of time. A Moving Average is most often used to average values for a smoother representation of the underlying price or indicator.	CLOSE > SMA(CLOSE, 30) Evaluates to true when the close is greater than a 30-day SMA.	Many traders watch for short-term averages to cross above longer-term averages to signal the beginning of an uptrend. Short-term averages (e.g. 15-period SMA) act as levels of support when the price experiences a pullback. Support levels become stronger and more

	Indicator	Signature	Description	Sample script	Remarks
					significant as the number of time periods used in the calculations increases.
2	Exponential Moving Average	EMA(Vector, Periods)	An Exponential Moving Average is similar to a Simple Moving Average. An EMA is calculated by applying a small percentage of the current value to the previous value, therefore an EMA applies more weight to recent values. A Moving Average is most often used to average values for a	CLOSE > EMA(CLOSE, 30) Evaluates to true when the close is greater than a 30-day EMA.	This type of moving average reacts faster to recent price changes than a simple moving average. The 12- and 26-day EMAs are the most popular short-term averages, and they are used to create indicators like the moving average convergence

	Indicator	Signature	Description	Sample script	Remarks
			smoother representation of the underlying price or indicator.		divergence (MACD) and the percentage price oscillator (PPO). In general, the 50- and 200-day EMAs are used as signals of long-term trends
3	Time Series Moving Average	TSMA(Vector, Periods)	A Time Series Moving Average is similar to a Simple Moving Average, except that values are derived from linear regression forecast values instead of regular values. A Moving Average is most often used to average values for a smoother representation of the underlying price or indicator.	CLOSE > TSMA(CLOSE, 30) Evaluates to true when the close is greater than a 30-day TSMA.	The time series moving average differs greatly from other types of moving averages in that the current value follows the recent trend of the data, not an actual average of the data. Because of this, the value of this function can be greater or less than all of the values being used if the

	Indicator	Signature	Description	Sample script	Remarks
					trend of the data is generally increasing or decreasing.
4	Variable Moving Average	VMA(Vector, Periods)	A Variable Moving Average is similar to an exponential moving average except that it adjusts to volatility. A Moving Average is most often used to average values for a smoother representation of the underlying price or indicator.	CLOSE > VMA(CLOSE, 30) Evaluates to true when the close is greater than a 30-day VMA.	A **VMA** is an EMA that's able to regulate its smoothing percentage based on market inconstancy automatically. Its sensitivity grows by providing more weight to the ongoing data as it generates a better signal indicator for short and long term markets. A Variable Moving Average regulates its sensitivity and lets it function better in any market

	Indicator	Signature	Description	Sample script	Remarks
					conditions by using automatic regulation of the smoothing constant.
5	Triangular Moving Average	TMA(Vector, Periods)	The Triangular Moving Average is similar to a Simple Moving Average, except that more weight is given to the price in the middle of the moving average periods. A Moving Average is most often used to average values for a smoother representation of the underlying price or indicator.	CLOSE > TMA(CLOSE, 30) Evaluates to true when the close is greater than a 30-day TMA.	As with other moving averages, the triangular moving average can be used to identify a trend by using the slope of the average (or lack of slope in a ranging market). However, due to the additional smoothing, triangular moving averages tend to be smoother, and have more waves, than standard moving

	Indicator	Signature	Description	Sample script	Remarks
					averages. Interestingly, triangular moving averages often appear more responsive to direction changes, even though the additional smoothing actually moves the dominant input value to the middle of the input series (which would decrease responsiveness).
6	Weighted Moving Average	WMA(Vector, Periods)	A Weighted Moving Average places more weight on recent values and less weight on older values. A Moving Average is most often used to	CLOSE > WMA(CLOSE, 30) Evaluates to true when the close is greater than a 30-day WMA.	The linearly weighted moving average was one of the first responses to placing a greater importance on recent data.

	Indicator	Signature	Description	Sample script	Remarks
			average values for a smoother representation of the underlying price or indicator.		The popularity of this moving average has been diminished by the exponential moving average, but none the less it still proves to be very useful.
7	WellesWilder Smoothing	WWS(Vector, Periods)	The Welles Wilder's Smoothing indicator is similar to an exponential moving average. The indicator does not use the standard exponential moving average formula. Welles Wilder described 1/14 of today's value + 13/14 of yesterday's average as a 14-day exponential moving average. This indicator	CLOSE > WWS(CLOSE, 30) Evaluates to true when the close is greater than a 30-day WWS.	It is slow to reflect price changes compared with other moving averages. Wilder's Smoothing is used as a part of Wilder's RSI.

	Indicator	Signature	Description	Sample script	Remarks
			is used in the manner that any other moving average would be used.		
8	VIDYA (Volatility Index Dynamic Average)	VIDYA(Vector, Periods, R2Scale)	VIDYA (Volatility Index Dynamic Average), developed by Mr. Tuschar Chande, is a moving average derived from linear regression R2.A Moving Average is most often used to average values for a smoother representation of the underlying price or indicator. Because VIDYA is a derivative of linear regression, it quickly adapts to volatility. Parameters R2Scale is a value	CLOSE > VIDYA(CLOSE, 30, 0.65) Evaluates to true when the close is greater than a 30-day VIDYA with an R2 of 0.65.	The VIDYA exhibits a trend-following character, as do other moving averages. For this reason, interpretations similar to those used for the moving averages can also be applied here.

	Indicator	Signature	Description	Sample script	Remarks
			specifying the R-Squared scale to use in the linear regression calculations. Mr. Chande recommends a value between 0.5 and 0.8 (default value is 0.65).		
9	R-squared	R2(Vector, Periods)	R-Squared is the coefficient of determination for the supplied vector over the specified periods. The values oscillate between 0 and 1.	R2(CLOSE, 30) < 0.1 Evaluates to true when the coefficient of determination is less than 0.1.	R-squared shows the strength of trend. The more closely prices move in a linear relationship with the passing of time, the stronger the trend. When using momentum based indicators, only trade overbought/oversold levels if you have determined that prices are trendless

	Indicator	Signature	Description	Sample script	Remarks
					or weakening (i.e., a low or lowering r-squared value). Because in a strong trending market, prices can remain overbought or oversold for extended periods
10	Slope	Slope(Vector, Periods)	Returns the linear regression slope value for the data being analyzed over the specified number of periods. Values oscillate from negative to positive numbers.	SLOPE(CLOSE, 30) > 0.3 Evaluates to true when the slope is greater than 0.3.	If a price trend rises or descends, the linear regression slope suggests the possible angle of an uptrend/downtrend basing on the current price. It is thought that if the price differs from the linear regression line, it gets too strained and starts motion in

	Indicator	Signature	Description	Sample script	Remarks
					direction to the line. Thereby this monitor allows us to understand the moment of price trend change.
11	Forecast	Forecast(Vector, Periods)	Returns the linear regression forecast for the next period based on the linear regression calculation over the specified number of periods.	Forecast(CLOSE, 30) > REF(CLOSE,1) Evaluates to true when the forecast is higher than the previous closing price.	Forecast defines the trend's upward or downward declivity and stretches those results into the future. For instance, when prices are moving upwards, TSF tries to define the upward declivity of the price compared to the ongoing price and stretch that calculation forward. a lot of analysts think that once

	Indicator	Signature	Description	Sample script	Remarks
					prices shift above or fall below the indicator line; prices will likely move back to the line. The TSF indicator also defines if a change in direction happened monitoring the ongoing trend.
12	Intercept	Intercept(Vector, Periods)	Returns the linear regression intercept for the last period's Y value, based on the linear regression calculation over the specified number of periods.	Intercept(CLOSE, 30) > REF(CLOSE,1) Evaluates to true when the intercept is higher than the previous closing price.	Interpretations similar to those used for the Slope and forecast can also be applied here.
13	BollingerBandsTop	BollingerBandsTop(Vector, Periods, Standard Deviations,	Bollinger bands rely on standard deviations in order to adjust to	CLOSE > BBT(CLOSE, 20, 2, EXPONENTIAL)	The following traits are particular to the Bollinger Band:

	Indicator	Signature	Description	Sample script	Remarks
		MA Type)	changing market conditions. When a stock becomes volatile the bands widen (move further away from the average). Conversely, when the market becomes less volatile the bands contract (move closer to the average). Tightening of the bands is often used as an early indication that the stock's volatility is about to increase.Bollinger Bands (as with most bands) can be imposed over an actual price or another indicator. When prices rise above the upper band or fall below the lower band, a change in	Evaluates to true when the close is greater than a 20-day Bollinger Band Top calculated by 2 standard deviations, using an exponential moving average.	1. abrupt changes in prices tend to happen after the band has contracted due to decrease of volatility. 2. if prices break through the upper band, a continuation of the current trend is to be expected. 3. if the pikes and hollows outside the band are

	Indicator	Signature	Description	Sample script	Remarks
			direction may occur when the price penetrates the band after a small reversal from the opposite direction. Recommended Parameters Vector: CLOSE Periods: 20 Standard Deviations: 2 MA Type: EXPONENTIAL		followed by pikes and hollows inside the band, a reverse of trend may occur. 4. the price movement that has started from one of the band's lines usually reaches the opposite one. The last observation is useful for forecasting price guideposts.

	Indicator	Signature	Description	Sample script	Remarks
14	BollingerBands Middle	BollingerBandsMiddle(Vector, Periods, Standard Deviations, MA Type)	Refer Point 13.	CLOSE > BBM(CLOSE, 20, 2, EXPONENTIAL) Evaluates to true when the close is greater than a 20-day Bollinger Band Middle calculated by 2 standard deviations, using an exponential moving average.	Refer Point 13.
15	MovingAverage EnvelopeTop	MovingAverageEnvelopeTop(Periods, MA Type, Shift)	Moving Average Envelopes consist of moving averages calculated from the underling price, shifted up and down by a fixed percentage. Moving Average Envelopes (or trading bands) can be imposed	CLOSE > MAET(20, SIMPLE, 5) Evaluates to true when the close is greater than a 20-day Moving Average Envelope Top calculated by 5%	When prices rise above the upper band or fall below the lower band, a change in direction may occur when the price penetrates the band after a small reversal from the

	Indicator	Signature	Description	Sample script	Remarks
			over an actual price or another indicator. Recommended Parameters Periods: 20 MA Type: SIMPLE Shift: 5	using a simple moving average.	opposite direction.
16	MovingAverage EnvelopeBottom	MovingAverageEnvelopeBottom(Periods, MA Type, Shift)	Refer Point 15.	CLOSE > MAEB(20, SIMPLE, 5) Evaluates to true when the close is greater than a 20-day Moving Average Envelope Bottom calculated by 5% using a simple moving average.	Refer Point 15
17	PrimeNumberBandsTop	PrimeNumberBandsTop()	This novel indicator identifies the nearest prime number for the high and low and plots the two series as	CLOSE > PNBT() Evaluates to true when the close is greater than the Prime Number	NA

	Indicator	Signature	Description	Sample script	Remarks
			bands.	Bands Top.	
18	PrimeNumberBandsBottom	PrimeNumberBandsBottom()	Refer Point 17.	CLOSE > PNBB() Evaluates to true when the close is greater than the Prime Number Bands Top.	NA
19	MomentumOscillator	MomentumOscillator (Vector, Periods)	The momentum oscillator calculates the change of price over a specified length of time as a ratio.Increasingly high values of the momentum oscillator may indicate that prices are trending strongly upwards. The momentum oscillator is closely related to MACD and Price Rate of Change (ROC). Recommended Parameters	MO(CLOSE, 14) > 90 Evaluates to true when the momentum oscillator of the close is over 90.	Many leading indicators come in the form of momentum oscillators. Generally speaking, momentum measures the rate-of-change of a security's price. As the price of a security rises, price momentum increases. The faster the security rises (the greater the period-over-

	Indicator	Signature	Description	Sample script	Remarks
			Vector: CLOSE Periods: 14		period price change), the larger the increase in momentum. Once this rise begins to slow, momentum will also slow. As a security begins to trade flat, momentum starts to actually decline from previous high levels. However, declining momentum in the face of sideways trading is not always a bearish signal. It simply means that momentum is returning to a more median level.
20	ChandeMoment	ChandeMomentumO	The Chande	CMO(CLOSE, 14)	The security is

	Indicator	Signature	Description	Sample script	Remarks
	umOscillator	scillator(Vector, Periods)	Momentum Oscillator (CMO), developed by Tushar Chande, is an advanced momentum oscillator derived from linear regression. This indicator was published in his book titled "New Concepts in Technical Trading" in the mid 90's. The CMO enters into overbought territory at +50, and oversold territory at -50. You can also create buy/sell triggers based on a moving average of the CMO. Also, increasingly high values of CMO may indicate that prices are trending strongly upwards. Conversely, increasingly low values	> 48 Evaluates to true when the CMO of the close is overbought.	deemed to be overbought when the momentum oscillator is above +50 and oversold when it is below -50. Many technical traders add a nine-period moving average to this oscillator to act as a signal line. Bullish signals are generated when the oscillator crosses above the signal, and bearish signals are generated when the oscillator crosses down through the signal.

	Indicator	Signature	Description	Sample script	Remarks
			of CMO may indicate that prices are trending strongly downwards. Recommended Parameters Vector: CLOSE Periods: 14		
21	VolumeOscillator	VolumeOscillator(Short Term Periods, Long Term Periods, MA Type, Points or Percent)	The Volume Oscillator shows a spread of two different moving averages of volume over a specified period of time. Offers a clear view of whether or not volume is increasing or decreasing. Recommended Parameters Short Term Periods: 9 Long Term Periods: 21 MA Type: SIMPLE Points or Percent: PERCENT	VO(9, 21, SIMPLE, PERCENT) > 0	The Volume Oscillator identifies whether the volume trend is increasing or decreasing by using the difference between a longer and a shorter moving average of volume. When the Volume Oscillator rises above zero the shorter-term volume moving

	Indicator	Signature	Description	Sample script	Remarks
					average has risen above the longer-term volume moving average. This means that the short-term volume trend is higher than the longer-term volume trend. Rising prices along with increased volume, and falling prices along with decreased volume are interpreted to be bullish signals. Conversely, volume increasing as prices fall, or volume decreases as prices rise, is interpreted as a bearish signal.
22	PriceOscillator	PriceOscillator(Vecto	Similar to the Volume	PO(CLOSE, 9, 14,	This indicator is a

	Indicator	Signature	Description	Sample script	Remarks
		r, Short Term Periods, Long Term Periods, MA Type)	Oscillator, the Price Oscillator is calculated based on a spread of two moving averages.The Price Oscillator is basically a moving average spread. Buying usually occurs when the oscillator rises, and selling usually occurs when the oscillator falls. Recommended Parameters Vector: CLOSE Short Term Periods: 9 Long Term Periods: 14 MA Type: SIMPLE	SIMPLE) > 0 Evaluates to true when the Price Oscillator is in positive territory.	trend following system, majority of traders follow a very simple rule of buying when the shorter average crosses above the longer average and conversely when the shorter average crosses below the longer average a sell signal is triggered. Another method is to fade the signals and go in the opposite direction. This generally works better in choppy markets, as the moving averages are not permitted to trend due to a range bound

	Indicator	Signature	Description	Sample script	Remarks
					market.
23	DetrendedPrice Oscillator	DetrendedPriceOscillator(Vector, Periods, MA Type)	Similar to the Price Oscillator except DPO is used when long-term trends or outliers make the underlying price difficult to analyze. Buying occurs when the oscillator rises. Selling occurs when the oscillator falls. Recommended Parameters Vector: CLOSE Periods: 20 MA Type: SIMPLE	DPO(CLOSE, 20, SIMPLE) > 0 Evaluates to true when the Detrended Price Oscillator is in positive territory.	When the Detrended Price Oscillator is above the zero line, it means that price is above its moving average, a bullish sign. Similarly, when the Detrended Price Oscillator is below the zero line, it means that price is below its moving average, a bearish sign. The Detrended Price Oscillator is an effective tool for uncovering hidden cycles of overbought and oversold conditions.

	Indicator	Signature	Description	Sample script	Remarks
24	PrimeNumberOscillator	PrimeNumberOscillator(Vector)	Finds the nearest prime number from either the top or bottom of the series, and plots the difference between that prime number and the series itself. Recommended Parameters Vector: CLOSE	PNO(CLOSE) = REF(PNO(CLOSE), 1) AND REF(PNO(CLOSE), 2) != PNO(CLOSE)	This indicator can be used to spot market turning points. When the oscillator remains at the same high point for two consecutive periods in the positive range, consider selling. Conversely, when the oscillator remains at a low point for two consecutive periods in the negative range, consider buying.
25	FractalChaosOscillator	FractalChaosOscillator(Periods)	A buy signal is generated when the oscillator tops, and a sell signal is generated when the oscillator bottoms.	FCO(21) > REF(FCO(21),1)	The chaotic nature of stock market movements explains why it is sometimes difficult to distinguish daily

	Indicator	Signature	Description	Sample script	Remarks
			Recommended Parameters Periods: 21		charts from monthly charts if the time scale is not given. The patterns are similar regardless of the time resolution. Like the chambers of the nautilus, each level is like the one before it, but the size is different. To determine what is happening in the current level of resolution, the fractal chaos oscillator can be used to examine these patterns.
26	RainbowOscillator	RainbowOscillator(Vector, Levels, MA Type)	The rainbow oscillator is calculated based upon multiple time frames of a moving	SET R = RBO(CLOSE, 3, SIMPLE) R > 0.8 AND	When the market is rising and the trend is up, the least smoothed line is at

	Indicator	Signature	Description	Sample script	Remarks
			average. The trend may reverse suddenly when values stay above 0.80 or below 0.20 for two consecutive days. Recommended Parameters Vector: CLOSE Levels: 3 MA Type: SIMPLE	REF(R, 1) > 0.8 Evaluates to true when the Rainbow Oscillator has been above 0.8 for at least two consecutive days.	the top of the Rainbow (the red line) and the most smoothed line is at the bottom of the Rainbow (the violet line). When the market is declining and the trend is down the order of the Rainbow is reversed; the most smoothed line is at the bottom and the least smoothed line is at the top. As the underlying market price moves up or down, the moving averages follow and cross in sequence as the move continues.

	Indicator	Signature	Description	Sample script	Remarks
					Price moves away from the Rainbow are seen as potential trend continuations leading to a greater Rainbow width, while price moves towards (or into) the Rainbow are seen as potential reversals leading to a contraction of the Rainbow width. The depth that price penetrates into the Rainbow can be used to judge the strength of the move.
27	TRIX	TRIX(Vector, Periods)	RIX is a momentum oscillator that shows the rate of change of	TRIX(CLOSE, 9) > 0.9 Evaluates to true	The TRIX can help identify turning points. The simplest

	Indicator	Signature	Description	Sample script	Remarks
			an exponentially averaged closing price. Interpretation The most common usage of the TRIX oscillator is to buy when the oscillator rises and sell when the oscillator falls. Recommended Parameters Vector: CLOSE Periods: 9	when TRIX is in overbought territory.	rule of trading decision making while following the trend is to buy, when TRIX changes direction from decreasing to increasing. And to sell, when TRIX changes direction from increasing to decreasing
28	VerticalHorizontalFilter	VerticalHorizontalFilter(Vector, Periods)	The Vertical Horizontal Filter (VHF) identifies whether a market is in a trending or a choppy movement phase.The VHF indicator is most commonly used as an indicator of market volatility. It is also frequently used as a component to other	VHF(CLOSE, 21) < 0.2 Evaluates to true if the vertical horizontal filter for the closing price of last 21 days is less than 0.2.A value higher than 0.2 signifies a start of	The VHF indicator determines the trend of prices to help to decide which indicators to use. The higher the VHF, the higher the degree of trending and the more you should be using trend- following

	Indicator	Signature	Description	Sample script	Remarks
			technical indicators. Recommended Parameters Vector: CLOSE Periods: 21	an uptrend.	indicators. Rising VHF indicates a developing trend; falling VHF indicates that prices may be entering a congestion phase. Vertical Horizontal Filter does not, itself, generate trading signals, but determines whether signals are taken from trend or momentum indicators.
29	EaseOfMovement	EaseOfMovement(Vector, Periods)	The Ease of Movement oscillator displays a unique relationship between price change and volume. The Ease of Movement	EOM(CLOSE, 21) > 0 Evaluates to true when the Ease of Movement is in positive territory	It highlights the relationship between volume and price changes and is particularly useful for assessing

	Indicator	Signature	Description	Sample script	Remarks
			oscillator rises when prices are trending upwards under low volume, and likewise, the Ease of Movement oscillator falls when prices are trending downwards under low volume. Recommended Parameters Vector: CLOSE Periods: 21		the strength of a trend. The Ease of Movement indicator produces a buy signal when it crosses above zero, indicating that prices are moving upward more easily; a sell signal is given when the indicator crosses below zero, indicating that prices are moving downward more easily.
30	Wilder's directional movement system	ADX(Periods), ADXR(Periods), DIP(Periods), DIN(Periods), TRSUM(Periods), DX(Periods)	The Welles Wilder's Directional Movement System contains five indicators; ADX, DI+, DI-, DX, and ADXR. A buy signal is given when DI+ crosses over	DIP(14) > 60 This evaluates to true if the DI+ for 14 ticks is more than 60.This summarizes an uptrend in	The ADX (Average Directional Movement Index) is an indicator of how much the market is trending, either up or down: the higher

	Indicator	Signature	Description	Sample script	Remarks
			DI-, a sell signal is given when Di- crosses over DI+. Recommended Parameters Periods: 21	progress.	the ADX line, the more the market is trending and the more suitable it becomes for a trend-following system. This indicator consists of two lines: DI+ and DI-, the first one being a measure of uptrend and the second one a measure of downtrend. Detailed information about this indicator and formulas can be found in Welles Wilder's book, "New Concepts in Technical Trading Systems". The standard

	Indicator	Signature	Description	Sample script	Remarks
					Directional Movement System draws a 14 period DI+ and a 14 period DI- in the same chart panel. ADX is also sometimes shown in the same chart panel.
31	Williams % R	WilliamsPctR(Periods)	Developed by trader Larry Williams, the Williams' %R indicator measures overbought/oversold levels. This indicator is similar to the Stochastic Oscillator. The outputs range from 0 to -100. Interpretation The market is considered overbought when the %R is in a range of 0 to	WPR(14) < -80 Evaluates to true when Williams' %R is oversold.	Williams's %R has proven very useful for anticipating market reversals. It identifies overbought or oversold markets. It is important to remember that overbought does not necessarily imply time to sell and oversold does not necessarily imply time to buy. A

	Indicator	Signature	Description	Sample script	Remarks
			-20, and oversold when %R is in a range of -80 to -100. Recommended Parameters Periods: 14		security can be in a downtrend, become oversold and remain oversold as the price continues to trend lower. Once a security becomes overbought or oversold, traders should wait for a signal that a price reversal has occurred. One method might be to wait for Williams %R to cross above or below -50 for confirmation. Price reversal confirmation can also be accomplished by using other indicators or

	Indicator	Signature	Description	Sample script	Remarks
					aspects of technical analysis in conjunction with Williams %R.
32	Williams Accumulation/Distribution	WilliamsAccumulationDistribution()	Another indicator developed by trader Larry Williams, the Accumulation / Distribution indicator shows a relationship of price and volume.When the indicator is rising, the security is said to be accumulating. Conversely, when the indicator is falling, the security is said to being distributing. Prices may reverse when the indicator converges with price.	WAD() < 1 Evaluates to true when Williams' Accumulation / Distribution is below 1.	The Williams Accumulation/Distribution Indicator tries to find underlying relationships between the close, high and low prices. It tracks the buying pressure and selling pressure. Williams Accumulation-Distribution (WAD) tracks buying pressure (accumulation) and selling pressure

	Indicator	Signature	Description	Sample script	Remarks
					(distribution) on a security. With accumulation, most of the volume is associated with upward price movement. With distribution, most of the volume is associated with downward price movement.
33	ChaikinVolatility	ChaikinVolatility(Periods, Rate of Change, MA Type)	The Chaikin Volatility Oscillator is a moving average derivative of the Accumulation / Distribution index. This indicator quantifies volatility as a widening of the range between the high and the low price.The Chaikin	CV(10, 10, SIMPLE) < -25	It measures volatility as the trading range between high and low for each period. There are two ways to interpret this measure of

	Indicator	Signature	Description	Sample script	Remarks
			Volatility Oscillator adjusts with respect to volatility, independent of long-term price action. The most popular interpretation is to sell when the indicator tops out, and to buy when the indicator bottoms out. Recommended Parameters Periods: 10 Rate of Change: 10 MA Type: SIMPLE		volatility. One method assumes that market tops are generally accompanied by increased volatility and that the latter stages of a market bottom are generally accompanied by decreased volatility. Another method assumes that an increase in the Volatility indicator over a relatively short time period indicates that a bottom is near (e.g.,

	Indicator	Signature	Description	Sample script	Remarks
					a panic sell-off) and that a decrease in volatility over a longer time period indicates an approaching top (e.g., a mature bull market).
34	Aroon	AroonUp(Periods) AroonDown(Periods)	The Aroon indicator was developed by Tushar Chande in the mid 1990's. This indicator is often used to determine whether a stock is trending or not and how stable the trend is.Trends are determined by extreme values (above 80) of both lines (Aroon up and Aroon down), whereas unstable prices are	AroonUp(25) > 80 AND AroonDown(25) > 80	Aroon is used to measure the presence and strength of trends. When Aroon(up) and Aroon(down) are moving together, there is no clear trend (the price is moving sideways, or about to move sideways). When the Aroon(up) is below 50, it is an

	Indicator	Signature	Description	Sample script	Remarks
			determined when both lines are low (less than 20). Recommended Parameters Periods: 25		indication that the uptrend is losing its momentum, while when the Aroon(down) is below 50, it is an indication that the downtrend is losing its momentum. When the Aroon(up) or Aroon(down) are above 70, it indicate a strong trend in the same direction, while when the value is below 30, it indicates a trend coming in an opposite direction. Finally, for the Aroon Oscillator, the positive value indicates an upward trend (or coming

	Indicator	Signature	Description	Sample script	Remarks
					trend), and the negative value indicates a downward trend. The higher the absolute value of an oscillator, the stronger is an indication of a trend.
35	Moving Average Convergence/Divergence(MACD)	MACD(Short Cycle, Long Cycle, Signal Periods, MA Type)	The MACD is a moving average oscillator that shows potential overbought/oversold phases of market fluctuation. The MACD is a calculation of two moving averages of the underlying price/indicator. Buy and sell interpretations may be derived from crossovers (calculated	SET A = MACDSignal(13, 26, 9, SIMPLE) SET B = MACD(13, 26, 9, SIMPLE) CROSSOVER(A, B) = TRUE Evaluates to true when the MACD Signal line recently crossed over the MACD.	The MACD proves most effective in wide-swinging trading markets. The basic MACD trading rule is to sell when the MACD falls below its signal line. Similarly, a buy signal occurs when the MACD rises above its signal line. It is also popular to buy/sell when the

	Indicator	Signature	Description	Sample script	Remarks
			by the MACDSignal function), overbought / oversold levels of the MACD and divergences between MACD and underlying price. Recommended Parameters Long Cycle: 26 Short Cycle: 13 Signal Periods: 9 MA Type: SIMPLE		MACD goes above/below zero. The MACD is also useful as an overbought/oversold indicator.
36	HighMinusLow	HighMinusLow()	This function returns the high price minus the low price for each bar.This indicator is often used as a component for other technical indicators but can be used with a moving average to show the change in price action over time.	SET A = SMA(HML(), 14) A > REF(A, 10) Evaluates to true when the height of each bar has been increasing over the past several bars.	NA

	Indicator	Signature	Description	Sample script	Remarks
37	Stochastic Oscillator	SOPK(%K Periods, %K Slowing Periods, %D Periods, MA Type)	The Stochastic Oscillator is a popular indicator that shows where a security's price has closed in proportion to its closing price range over a specified period of time. Interpretation The Stochastic Oscillator has two components: %K (the SOPK function) and %D (the SOPD function). %K is most often displayed on a stock chart as a solid line and %D is often shown as a dotted line. The most widely used method for interpreting the Stochastic Oscillator is to buy when either	SOPK(9, 3, 9, SIMPLE) > 80 OR SOPD(9, 3, 9, SIMPLE) > 80 Evaluates to true when the Stochastic Oscillator is in oversold territory.	The stochastic indicator is: • a momentum oscillator that can warn of strength or weakness in the market, often well ahead of the final turning point. • based on the assump

	Indicator	Signature	Description	Sample script	Remarks
			component rises above 80 or sell when either component falls below 20. Another way to interpret the Stochastic Oscillator is to buy when %K rises above %D, and conversely, sell when %K falls below %D. Recommended Parameters % K Periods: 9 % K Slowing Periods: 3 % D Periods: 9 MA Type: SIMPLE		tion that when an instrum ent is rising it tends to close near the high and when an instrum ent is falling it tends to close near its lows.
38	RelativeStrength Index	RelativeStrengthInde x(Vector, Periods)	The RSI is popular indicator developed by	RSI(CLOSE, 14) > 55	The Relative Strength Index

	Indicator	Signature	Description	Sample script	Remarks
			trader Welles Wilder. The RSI is a popular indicator that shows comparative price strength within a single security. Interpretation The most widely used method for interpreting the RSI is price / RSI divergence, support / resistance levels and RSI chart formations. Recommended Parameters Vector: CLOSE Periods: 14		compares upward movements in closing price to downward movements over a selected period. Wilder recommended using 70 and 30 and overbought and oversold levels respectively. Generally, if the RSI rises above 30 it is considered bullish for the underlying stock. Conversely, if the RSI falls below 70, it is a bearish signal. Some traders identify the long-term trend and then use extreme readings for entry points. If

	Indicator	Signature	Description	Sample script	Remarks
					the long-term trend is bullish, then oversold readings could mark potential entry points.
39	MassIndex	MassIndex(Periods)	The Mass Index identifies price changes by indexing the narrowing and widening change between high and low prices. Interpretation According to the inventor of the Mass Index, reversals may occur when a 25-period Mass Index rises above 27 or falls below 26.5. Recommended Parameters Periods: 25	MI(25) > 27	It is used to identify trend reversals. The Mass Index is a range oscillator that uses changes in trading price and provides unique market reversal forecasts that other indicators may miss. The Mass Index attempts to identify reversals by comparing the trading range between High & low prices for each period. A bulge in

	Indicator	Signature	Description	Sample script	Remarks
					the index line signals reversals. This occurs when the 25 periods Mass Index rises above 27 and falls back below 26.5. By calculating a 9 days exponential moving average, go long if there is a reversal bulge & EMA points downwards, whereas go short if there is reversal bulge and EMA points upward
40	HistoricalVolatilityIndex	HistoricalVolatilityIndex(Vector, Periods, Bar History, Standard Deviations)	Historical volatility is the log-normal standard deviation. The Historical Volatility Index is based on the book by Don Fishback,	HVI(CLOSE, 15, 30, 2) < 0.01	High values of HVI indicate that the stock is volatile, while low values of HVI indicate that the stock is either

	Indicator	Signature	Description	Sample script	Remarks
			"Odds: The Key to 90% Winners". The formula for a 30-day historical volatility index between 1 and 0 is: Stdev(Log(Close / Close Yesterday), 30) * Sqrt(365) Some traders use 252 instead of 365 for the bar history that is used by the square root function. The Log value is a natural log (I.e. Log10). Recommended Parameters Vector: CLOSE Periods: 15 Bar History: 30 Standard Deviations: 2		flat or trending steadily.
41	ChaikinMoneyFlow	ChaikinMoneyFlow (Periods)	The Chaikin Money Flow oscillator is a	CMF(15) > 20 AND	The indicator often warns of break-outs

	Indicator	Signature	Description	Sample script	Remarks
			momentum indicator that spots buying and selling by calculating price and volume together. This indicator is based upon Accumulation / Distribution, which is in turn based upon the premise that if a stock closes above its midpoint, (high + low) / 2, for the day then there was accumulation that day, and if it closes below its midpoint, then there was distribution that day. Interpretation A buy signal is generated when the indicator is rising and is in positive territory. A sell signal is	REF(CMF(15), 1) > 20 Evaluates to true when the Chaikin Money Flow Index is bullish.	& provides useful trend confirmation. When price is trading in a range, volume may indicate in which direction a breakout is most likely to occur. Higher volume at peaks means that an upward breakout is more likely. Higher volume at troughs indicates that a downward breakout is more likely.

	Indicator	Signature	Description	Sample script	Remarks
			generated when the indicator is falling and is in negative territory. Recommended Parameters Periods: 15		
42	ComparativeRelativeStrength	ComparativeRelativeStrength(Vector1, Vector2)	The Comparative Relative Strength index compares one vector with another. Interpretation The base vector is outperforming the other vector when the Comparative RSI is trending upwards. Recommended Parameters	CRSI(CLOSE, VOLUME) > 1 Evaluates to true when the trend in price has outpaced the trend in volume.	Comparative Relative Strength is often used to compare a security's performance with a market index. It is also useful in developing spreads (i.e., buy the best performer and short the weaker issue).
			Vector1: CLOSE Vector2: [Any]		
43	PriceVolumeTrend	PriceVolumeTrend(Vector)	Also known as *Volume Price Trend*. This	TREND(PVT(CLOSE)) = UP	It used to determine the

	Indicator	Signature	Description	Sample script	Remarks
			indicator consists of a cumulative volume that adds or subtracts a multiple of the percentage change in price trend and current volume, depending upon their upward or downward movements. PVT is used to determine the balance between a stock's demand and supply. This indicator shares similarities with the On Balance Volume index. Interpretation The Price and Volume Trend index generally precedes actual price movements. The premise is that well-informed investors are buying	Evaluates to true when PVT is trending upwards.	strength of trends and warn of reversals. PVT is a leading indicator for future price movements. PVT more accurately demonstrates the flow of money. The Price Volume Trend is an oscillator, which attempts to measure the strength of trends based upon volume and price change. The Price Volume Trend is used in conjunction with the price plot. Prices are expected to rise when there is a bullish divergence between the Price

	Indicator	Signature	Description	Sample script	Remarks
			when the index rises and uninformed investors are buying when the index falls. Recommended Parameters Vector: CLOSE		Volume Trend and the closing price and prices are expected to fall if there is a bearish divergence.
44	PositiveVolumeIndex	PositiveVolumeIndex (Vector)	The Positive Volume Index puts focus on periods when volume increases from the previous period. Interpretation The interpretation of the Positive Volume Index is that the majority of investors are buying when the index rises, and selling when the index falls. Recommended Parameters Vector: CLOSE	TREND(PVI(CLOSE)) = UP Evaluates to true when PVI is trending upwards.	It tries to determine what smart investors are doing. When trading volume is high it is thought that inexperienced investors are involved. Whereas on slow days, "shrewd investors" quietly buy or sell the stock.
45	NegativeVolume	NegativeVolumeInde	The Negative Volume	TREND(NVI(CLOS	The index tries to

	Indicator	Signature	Description	Sample script	Remarks
	Index	x(Vector)	Index is similar to the Positive Volume Index, except it puts focus on periods when volume *decreases* from the previous period. Interpretation The interpretation of the Negative Volume Index is that well-informed investors are buying when the index falls and uninformed investors are buying when the index rises. Recommended Parameters Vector: CLOSE	E)) = UP Evaluates to true when NVI is trending upwards.	determine what smart investors are doing. It is believed that when volume is high, uninformed investors will sell. While on slow days, "shrewd investors" will quietly buy or sell the stock.
46	OnBalanceVolume	OnBalanceVolume(Vector)	The On Balance Volume index shows a relationship of price and volume in the form of a momentum index.	TREND(OBV(CLOSE)) = UP Evaluates to true when OBV is trending upwards.	The basic assumption, regarding OBV analysis, is that OBV changes precede price changes. The

	Indicator	Signature	Description	Sample script	Remarks
			Interpretation On Balance Volume generally precedes actual price movements. The premise is that well-informed investors are buying when the index rises and uninformed investors are buying when the index falls. Recommended Parameters Vector: CLOSE		theory is that smart money can be seen flowing into the instrument by a rising OBV. When the public then moves into the instrument, both the instrument and the OBV will surge ahead.
47	PerformanceIndex	PerformanceIndex(Vector)	The Performance indicator calculates price performance as a normalized value or percentage. Interpretation A Performance indicator shows the price of a security as a normalized value. If	PFI(CLOSE) > 45 Evaluates to true when the performance index is over 45% Trade Volume Index	Performance index are helpful for comparing the price movements of different securities

	Indicator	Signature	Description	Sample script	Remarks
			the Performance indicator shows 50, then the price of the underlying security has increased 50% since the start of the Performance indicator calculations. Conversely, if the indictor shows -50, then the price of the underlying security has decreased 50% since the start of the Performance indicator calculations. Recommended Parameters Vector: CLOSE		
48	TradeVolumeIndex	TradeVolumeIndex(Vector, Minimum Tick Value)	The Trade Volume index shows whether a security is being accumulated or distributed (similar to	TVI(CLOSE, 0.25) > 0 Evaluates to true when the Trade Volume Index is	The TVI helps identify whether buyers or sellers are in control. If the TVI is trending up, it

	Indicator	Signature	Description	Sample script	Remarks
			the Accumulation/Distribution index). Interpretation When the indicator is rising, the security is said to be accumulating. Conversely, when the indicator is falling, the security is said to being distributing. Prices may reverse when the indicator converges with price. Recommended Parameters Vector: CLOSE Minimum Tick Value: 0.25	in positive territory.-	indicates that buyers are in control. If the TVI is trending down, it indicates that sellers are in control. If the TVI is above zero, it indicates that net buying has taken place over the time period displayed. If the TVI is below zero, it indicates that net selling has taken place over the time period displayed
49	SwingIndex	SwingIndex(Limit Move Value)	The Swing Index (Wilder) is a popular indicator that shows comparative price	SI(1) > 0 Evaluates to true when the Swing Index is in	It provides an indication of the real strength and direction of a price

	Indicator	Signature	Description	Sample script	Remarks
			strength within a single security by comparing the current open, high, low, and close prices with previous prices. Interpretation The Swing Index is a component of the Accumulation Swing Index. Recommended Parameters Limit Move Value: 1	positive territory.	trend by providing a single indicator line, which can be analyzed for support and resistance lines
50	AccumulativeSwingIndex	AccumulativeSwingIndex(Limit Move Value)	The Accumulation Swing Index (Wilder) is a cumulative total of the Swing Index, which shows comparative price strength within a single security by comparing the current open, high, low, and close prices with	TREND(ASI(1)) > UP Evaluates to true when the Accumulative Swing Index is trending upwards.	It defines short-term swing points. It indicates the real strength and direction of the market. The accumulative swing index is used to gain a better long-term picture than

	Indicator	Signature	Description	Sample script	Remarks
			previous prices. Interpretation The Accumulation Swing Index may be analyzed using technical indicators, line studies, and chart patterns, as an alternative view of price action. Recommended Parameters Limit Move Value: 1		using the plain swing index, which uses data from only two bars. If the long-term trend is up, the accumulative swing index is a positive value. Conversely, if the long-term trend is down, the accumulative swing index is a negative value. If the long-term trend is sideways (non-trending), the accumulative swing index fluctuates between positive and negative values. This indicator is used to analyze futures but can be applied to

	Indicator	Signature	Description	Sample script	Remarks
					stocks as well.
51	CommodityChannelIndex	CommodityChannelIndex(Periods, MA Type)	Donald Lambert developed the CCI indicator. Although the purpose of this indicator is to identify cyclical turns in commodities, it is often used for securities. Interpretation This indicator oscillates between an overbought and oversold condition and works best in a sideways market. Recommended Parameters Periods: 21 MA Type: SIMPLE	CCI(12, SIMPLE) > 0 AND REF(CCI(12, SIMPLE), 1) < 0 Evaluates to true when the CCI has just moved into positive territory.	The CCI is a versatile indicator capable of producing a wide array of buy and sell signals. CCI can be used to identify overbought and oversold levels. A security would be deemed oversold when the CCI dips below -100 and overbought when it exceeds +100. From oversold levels, a buy signal might be given when the CCI moves back above -100. From overbought levels, a sell signal might be given when the CCI

	Indicator	Signature	Description	Sample script	Remarks
					moved back below +100. CCI also help identify price reversals, price extremes and trend strength. CCI fits into the momentum category of oscillators.
52	ParabolicSAR	ParabolicSAR(Min AF, Max AF)	Author Welles Wilder developed the Parabolic SAR. This indicator is always in the market (whenever a position is closed, an opposing position is taken). The Parabolic SAR indicator is most often used to set trailing price stops.	CROSSOVER(CLOSE, PSAR(0.02, 0.2)) = TRUE Evaluates to true when the close recently crossed over the Parabolic SAR.	The Parabolic SAR provides excellent exit points. You should close long positions when the price falls below the SAR and close short positions when the price rises above the SAR.

	Indicator	Signature	Description	Sample script	Remarks
			Interpretation A stop and reversal (SAR) occurs when the price penetrates a Parabolic SAR level. Recommended Parameters Min AF (Accumulation Factor): 0.02 Max AF (Accumulation Factor): 0.2		If long (i.e., the price is above the SAR), the SAR will move up every period, regardless of the direction the price is moving. The amount the SAR moves up depends on the amount that prices move. Parameters Minimum acceleration factor (Min AF): The values used for initializing and stepping the acceleration factor

	Indicator	Signature	Description	Sample script	Remarks
					(AF). It is also called initial/Incremental step. The default is set to .02 Maximum Acceleration factor (Max AF): - The maximum value that can be used for the acceleration factor (AF). The default is set to .2
53	Stochastic Momentum Index	SMIK(%K Periods, %K Smooth, %K Double Periods, %D Periods, MA Type, %D MA Type) SMID(%K Periods, %K Smooth, %K Double Periods, %D Periods, MA Type, %D MA	The Stochastic Momentum Index, developed by William Blau, first appeared in the January 1993 issue of Stocks & Commodities magazine. This indicator plots the	SMID(14, 2, 3, 9, SIMPLE, SIMPLE) > 40 OR SMIK(14, 2, 3, 9, SIMPLE, SIMPLE) > 40 Evaluates to true when the Stochastic	It is used as a sentiment, or trend identification indicator, thereby providing a better sense of the overall direction of the market. The interpretation of

	Indicator	Signature	Description	Sample script	Remarks
		Type)	closeness of price relative to the midpoint of the recent high / low range. Interpretation The Stochastic Momentum Index has two components: %K (SMIK) and %D (SMID). %K is most often displayed on a chart as a solid line and %D is often shown as a dotted line. The most widely used method for interpreting the Stochastic Momentum Index is to buy when either component rises above 40 or sell when either component falls below 40. Another way to interpret the Stochastic Momentum Index is to buy when	Momentum Index is in oversold territory.	the SMI is virtually identical to that of the Stochastic Oscillator. The most basic pattern to trade from is to buy when the SMI falls below -40 and then returns above it. Sell when the SMI rises above +40 and then falls back below that level. Another trading signal is buy when the SMI rises above the moving average, and sell when the SMI falls below the moving average.

	Indicator	Signature	Description	Sample script	Remarks
			%K rises above %D, or sell when %K falls below %D. Recommended Parameters %K Periods: 14 %K Smoothing: 2 %K Double Periods: 3 %D Periods: 9 MA Type: SIMPLE %D MA Type: SIMPLE		
54	MedianPrice	MedianPrice()	A Median Price is simply an average of one period's high and low values. Interpretation A Median Price is often used as an alternative way of viewing price action and also as a component for calculating other technical indicators.	CROSSOVER(CLOSE, SMA(MP(), 14)) Evaluates to true when the close crossed over the 14-day SMA of the Median Price.	A Median Price is often used as an alternative way of viewing price action, and also as a component for calculating other indicators.

	Indicator	Signature	Description	Sample script	Remarks
55	TypicalPrice	TypicalPrice()	A Typical Price is an average of one period's high, low, and close values. Interpretation A Typical Price is used as a component for the calculation of several technical indicators.	CROSSOVER(CLOSE, SMA(TP(), 14)) Evaluates to true when the close crossed over the 14-day SMA of the Typical Price.	For day trading, the Typical Price helps you to get a clear view of what the main thrust of the day's action were
56	WeightedClose	WeightedClose()	Weighted Close is an average of each day's open, high, low, and close, where more weight is placed on the close. Interpretation The Weighted Close indicator is a simple method that offers a simplistic view of market prices.	WC() > REF(WC(), 1) Evaluates to true when the weighted close is higher than the previous value.	The weighted close study is another way of viewing the price data. It places a greater emphasis on the closing price rather than the high or low. This process creates a single line chart. It provides clear and concise picture of the market
57	PriceRateOfCha	PriceRateOfChange(V	The Price ROC shows	PROC(CLOSE, 12)	The ROC displays

	Indicator	Signature	Description	Sample script	Remarks
	nge	ector, Periods)	the difference between the current price and the price one or more periods in the past. Interpretation A 12-day Price ROC is most often used as an overbought/oversold indicator. Recommended Parameters Vector: CLOSE Periods: 12	> 0 AND REF(PROC(CLOSE , 12),1) < 0 Evaluates to true when the Price ROC recently shifted into positive territory.	the wave-like motion in an oscillator format by measuring the amount that prices have changed over a given time period. This cyclical action is the result of the changing expectations as bulls and bears struggle to control prices.
58	VolumeRateOfChange	VolumeRateOfChange(Vector, Periods)	The Volume Rate of Change indicator shows whether or not volume is trending in one direction or another. Interpretation Sharp Volume ROC increases may signal price breakouts.	VROC(VOLUME, 12) > 0 AND REF(VROC(VOLUME, 12), 1) < 0 Evaluates to true when the Volume ROC recently moved into positive territory.	With most markets, the volume can be expected to within a constant range over time. When volume moves outside this range and begins to trend either upwards or downwards, then a

	Indicator	Signature	Description	Sample script	Remarks
			Recommended Parameters Vector: VOLUME Periods: 12		capitulation of one sort or another can be expected. Using this breakout from the average, the VROC is best used as a confirmation indicator to other studies. If the volume for the current period is higher than n-period ago, the rate of change will be a plus number. If volume is lower, the ROC will be minus number. This allows looking at the speed at which the volume is changing.

	Indicator	Signature	Description	Sample script	Remarks
59	HighestHighValue	HighestHighValue(Periods)	Returns the highest value of the high price over the specified number of periods. Interpretation Used as a component for calculation by many other technical indicators. Recommended Parameters Periods: 21	HIGH = HHV(21) Evaluates to true when the high is the highest high in the past 21 bars.	NA
60	LowestLowValue	LowestLowValue(Periods)	Returns the lowest value of the low price over the specified number of periods. Interpretation Used as a component for calculation by many other technical indicators. Recommended Parameters	LOW = LLV(21) Evaluates to true when the low is the lowest low in the past 21 bars.	NA

	Indicator	Signature	Description	Sample script	Remarks
			Periods: 21		
61	StandardDeviations	StandardDeviations(Vector, Periods, Standard Deviations, MA Type)	Standard Deviation is a common statistical calculation that measures volatility. Many technical indicators rely on standard deviations as part of their calculation. Interpretation Major highs and lows often accompany extreme volatility. High values of standard deviations indicate that the price or indicator is more volatile than usual. Recommended Parameters Vector: CLOSE Periods: 21 Standard Deviations: 2	SDV(CLOSE, 21, 2, SIMPLE) > REF(SDV(CLOSE, 21, 2, SIMPLE), 10) Evaluates to true when 21 period Standard Deviations are greater than 10 days ago.	It is used to measure the volatility. High values of standard deviations indicate that the price or indicator is more volatile than usual

	Indicator	Signature	Description	Sample script	Remarks
			MA Type: SIMPLE		
62	CorrelationAnalysis	CorrelationAnalysis(Vector1, Vector2)	Correlation analysis is used to determine the relationship between two vectors. Interpretation The function returns a value indicating the relationship between two Vectors. The Vectors may contain price, indicator values, or other values. Recommended Parameters Vector1: [Any Vector] Vector2: [Any Vector]	CA(CLOSE, SMA(CLOSE,14))> 0. Evaluates to true when the close price movement highly correlates with the 14- day SMA movement.	**Correlation Analysis** compares a stock to any indicator or another stock and demonstrates how like or dislike they are to one another. One can use correlation analysis in two main ways - for defining connection between two securities and for defining the ability of an indicator to forecast the situation on the market. Also correlation analysis is useful for measuring the

	Indicator	Signature	Description	Sample script	Remarks
					connection between two securities. As it often happens that one security's price causes or forecasts the price of another security. For instance, the correlation coefficient of gold against the dollar demonstrates a strong negative relationship. Thus, the dollar increase as a rule forecasts a decrease in the gold's price.