
August 2003

TRADERS' TIPS

Here is this month's selection of Traders' Tips, contributed by various developers of technical analysis software to help readers more easily implement some of the strategies presented in this and other issues.

You can copy these formulas and programs for easy use in your spreadsheet or analysis software. Simply select the desired text by highlighting as you would in any word processing program, then use your standard key command for copy or choose "copy" from the browser menu. The copied text can then be pasted into any open spreadsheet or other software by selecting an insertion point and executing a paste command. By toggling back and forth between an application window and the open Web page, data can be transferred with ease.

This month's tips include formulas and programs for:

- ▶ [TRADESTATION: REVERSE ENGINEERING RSI \(II\)](#)
- ▶ [TRADESTATION: DAYTRADING STRATEGIES](#)
- ▶ [METASTOCK: DAYTRADING STRATEGIES](#)
- ▶ [AMIBROKER: REVERSE ENGINEERING RSI \(II\)](#)
- ▶ [AMIBROKER: DAYTRADING STRATEGIES](#)
- ▶ [AIQ EXPERT DESIGN STUDIO: REVERSE ENGINEERING RSI \(II\)](#)
- ▶ [WEALTH-LAB: REVERSE ENGINEERING RSI \(II\)](#)
- ▶ [WEALTH-LAB: DAYTRADING STRATEGIES](#)
- ▶ [NEUROSHELL TRADER: DAYTRADING STRATEGIES](#)
- ▶ [NEOTICKER: REVERSE ENGINEERING RSI \(II\)](#)
- ▶ [NEOTICKER: DAYTRADING STRATEGIES](#)
- ▶ [TRADINGSOLUTIONS: REVERSE ENGINEERING RSI \(II\)](#)
- ▶ [INVESTOR/RT: DAYTRADING STRATEGIES](#)
- ▶ [STOCKWIZ: DAYTRADING STRATEGIES](#)
- ▶ [ASPEN RESEARCH: REVERSE ENGINEERING RSI \(II\)](#)
- ▶ [TECHNIFILTER PLUS: REVERSE ENGINEERING RSI \(II\)](#)

or return to [August 2003 Contents](#)

▶ [TRADESTATION: REVERSE ENGINEERING RSI \(II\)](#)

Giorgos Siligardos' article "Reverse Engineering RSI (II)" in this issue describes a translation process to plot the "next bar" price required to produce three variations on the relative strength index (RSI).



Figure1: TradeStation, reverse-engineered RSI. Here is the RevEngEMARSI plotted in TradeStation.

The first is an exponential moving average of the RSI. The second is a simple moving average of the RSI. The third is an RSItrendline. The EasyLanguage code for the last one, the RSI trendline, was presented in the June 2003 Traders' Tips. Here, we present the EasyLanguage code called "RevEngRSI-EMA" and "RevEngRSI-SMA" for exponential and simple moving averages.

Indicator: RevEngRSI-EMA

```
{RevEng EMARSI}
inputs:
    EmaPeriod( 45 ),
    WildPeriod( 14 );
variables:
    EmaRSI( 0 ),
    AUC(0),
    ADC(0),
    RevEngEmaRSI(0),
    X( 0 );
EmaRSI = XAverage( RSI( Close,WildPeriod ), EmaPeriod);
Value1 = Iff( Close > Close[1], Close - Close[1], 0 );
AUC = XAverage( Value1, WildPeriod );
Value2 = Iff( Close[1] > Close, Close[1] - Close, 0 );
ADC = XAverage( Value2, WildPeriod );
```

```

X = ( WildPeriod - 1 )*( ADC * EmaRSI
/ ( 100 - EmaRSI ) - AUC ) ;
RevEngEMARSI = Iff( X >= 0, Close + X,
Close + X * ( 100 - EmaRSI ) / EmaRSI ) ;
Plot1[-1]( RevEngEMARSI, "EmaRSI" ) ;

Indicator: RevEngRSI-SMA
{RevEng SMARSI}
Inputs:
    SmaPeriod( 45 ),
    WildPeriod( 14 );
variables:
    SmaRSI( 0 ),
    AUC( 0 ),
    ADC( 0 ),
    X( 0 ),
    RevEngSmaRSI( 0 );
SmaRSI = Average( RSI( Close, WildPeriod), SmaPeriod-1);
Value1 = Iff( Close > Close[1], Close - Close[1], 0 ) ;
AUC = XAverage( Value1, WildPeriod ) ;
Value2 = Iff( Close[1] > Close, Close[1] - Close, 0 ) ;
ADC = XAverage( Value2, WildPeriod ) ;
X = ( WildPeriod - 1 ) * ( ADC * SmaRSI
/ ( 100 - SmaRSI ) - AUC ) ;
RevEngSMARSI = Iff( X >= 0, Close + X,
Close + X * ( 100 - SmaRSI ) / SmaRSI ) ;
if CurrentBar > ( SmaPeriod ) then
    Plot1[-1] ( RevEngSMARSI, "SmaRSI" ) ;

```

The code and workspace will be available for download from the EasyLanguage Exchange at TradeStationWorld at <https://www.tradestationworld.com/discussions/default.asp?Group=8>. Look for the file "RevEng RSI 2.eld."

--Mark Mills
EasyLanguage Specialist
TradeStation Securities, Inc.
MarkM@TSSEC at EasyLanguage Questions Forum,
www.tradestation-support.com/discussions/forum.asp?FORUM_ID=184

[GO BACK](#)

▶TRADESTATION: DAYTRADING STRATEGIES

The EasyLanguage code given here reproduces the strategy calculations described in Jacob Singer's article, "Strategies For Daytrading." These calculations take the form of charts and spreadsheets. We have converted the spreadsheets into TradeStation RadarScreen pages.

RadarScreen is an integrated feature of TradeStation. Much like a "live spreadsheet," it allows you to define columns performing built-in or custom EasyLanguage code for real-time market scanning and ranking. Calculations can be for any time interval from one minute to one month, on current or historical data. Once calculated, the results can be sorted. In addition, the sort can be refreshed as often as every five seconds. Thresholds can be set to trigger audio alerts or change the color of the grid cell. Several hundred symbols can be loaded at a single time.

Singer calls his first strategy the pivot point. It consists of a symbol and 12 columns of "reports" on that symbol. We have converted the strategy into a RadarScreen page. The column calculations are done in three EasyLanguage documents, each producing four columns. "Perf, Vol & Pivot" includes the performance (net change) column -- upon which the entire RadarScreen is set to continuously report in real time. "!Stops & Tgt Levels," meanwhile, displays the author's key support and resistance prices; while "!BuySell & Alerts" displays the author's key buy/sell bias for the day, and the "bullish," "bearish," or "watch!" real-time setups and alerts.

```

Indicator: !Stops & Tgt Levels
{ !Stops & Tgt Levels
Jacob Singer, "DayTrading Strategies", TASC
Plots the author's stops & targets}
variables:
    FirstStop( 0 ),
    FirstTarget( 0 ),
    SecondStop( 0 ),
    SecondTarget( 0 ),
    TrueClose( 0 ) ;
{ Calculate the 'True Close' }
If High[1] = Close[1] then
    TrueClose = MaxList( High[1], Open )
Else if Low[1] = Close[1] then
    TrueClose = MinList( Low[1], Open )
Else TrueClose = Close[1] ;
{ Calculate Stops and Targets }
FirstStop = ( 2 * TrueClose ) - High ;

```

```

FirstTarget = ( 2 * TrueClose ) - Low ;
SecondStop = TrueClose - ( High - Low ) ;
SecondTarget = TrueClose + ( High - Low ) ;
Plot1( FirstStop, "1stStop" ) ;
Plot2( FirstTarget, "1stTarget" ) ;
Plot3( SecondStop, "2ndStop" ) ;
Plot4( SecondTarget, "2ndTarget" ) ;
Indicator: !BuySell and Alerts
{ !BuySell and Alerts
Jacob Singer, "DayTrading Strategies", TASC
Plots the author's BuySell & Alerts}
inputs:
  PerfLength( 104 ),
  VolLength( 25 ) ;
variables:
  BuySell( "" ),
  Perf( 0 ),
  VolAvg( 0 ),
  TrueClose( 0 ),
  BSColor( Green ),
  PPColor( Green ),
  PivotPoint( 0 ),
  PivotPointMA3( 0 ),
  PivotPointMA5( 0 ),
  Alert1( "" ),
  Alert2( "" ),
  Alert3( "" ),
  AlertString( "" );
{ Calculate 'True Close' }
if High[1] = Close[1] then
  TrueClose = MaxList( High[1], Open )
else if Low[1] = Close[1] then
  TrueClose = MinList( Low[1], Open )
else TrueClose = Close[1] ;
{ Calculate Buy/Sell Bias }
if Close - TrueClose > 0 then
  BuySell = "+1"
else if Close - TrueClose < 0 then
  BuySell = "-1"
else BuySell = "0" ;
{ Set Buy/Sell's Color ( Red or Green ) }
if BuySell = "+1" then BSColor = Green
else if BuySell = "-1" then BSColor = Red
else BSColor = Yellow ;
{ Calculate 'Watch' Alert }
VolAvg = AverageFC( Volume, VolLength ) ;
if ( Close[1] >= High[1] - .10 * Range[1] or
  Close[1] <= Low[1] + .10 * Range[1] ) and
  Volume[1] > ( 2 * VolAvg ) and
  Volume[1] < ( 4 * VolAvg ) then
begin
  Alert( "Watch!" ) ;
  Alert1 = " Watch! " ;
end
else Alert1 = "" ;
{ Calculate Pivot Point Alerts }
PivotPoint = ( High[1] + Low[1] + Close[1] ) / 3 ;
PivotPointMA3 = Average( PivotPoint, 3 ) ;
PivotPointMA5 = Average( PivotPoint, 5 ) ;
if Close[1] > PivotPoint and PivotPoint >
  PivotPointMA3 then
begin
  Alert( "Bullish Setup" ) ;
  Alert2 = " Bullish " ;
  PPColor = Green ;
end
else Alert2 = "" ;
if Close[1] < PivotPoint and PivotPoint <
  PivotPointMA3 and PivotPoint < PivotPointMA5 then
begin
  Alert( "Bearish Setup" ) ;
  Alert3 = " Bearish " ;
  PPColor = Red ;
end
else Alert3 = "" ;
AlertString = Alert1 + Alert2 + Alert3 ;
Plot1( BuySell, "Buy/Sell", BSColor ) ;
Plot2( AlertString, "Alert Status", PPColor ) ;
Indicator: !Perf, Vol & Pivot
{ !Perf, Vol & Pivot
Jacob Singer, "DayTrading Strategies", TASC
Plots the Performance (net gain) since x-days ago;
Volume; and pivot point (High + Low + Close)/3 }
inputs:
  PerfLength( 104 ),
  VolLength( 25 ) ;
variables:
  Perf( 0 ),
  PivotPoint( 0 ),
  PerfColor( Green ) ;

```

```

{ Calculate Performance (Net change since
'PerfLength' Days Ago) }
Perf = 100 * ( Close - Close[ PerfLength ] ) /
Close[ PerfLength ] ;
{ Set Perf's Color ( Red or Green ) }
if Perf > 0 then PerfColor = Green
else PerfColor = Red ;
{ Calculate Pivot }
PivotPoint = ( High[1] + Low[1] + Close[1] ) / 3 ;
Plot1( Perf, "Perf", PerfColor ) ;
Plot2( Volume, "Volume" ) ;
Plot3( PivotPoint, "Pivot" ) ;
The red-green strategy is Singer's next contribution.
We have coded it in EasyLanguage as follows:
Red-Green indicator:
Plot1[-10](Average(Average(H, 2), 2), "MA-Highs");
Plot2[-6](Average(Average(L, 2), 2), "MA-Lows");
Plot3(C, "Close");
Plot4(Average(C, 5), "MA Close");

```

The high-low strategy is Singer's last contribution. This strategy is another spreadsheet that can be easily translated into a TradeStation RadarScreen (Figure 2). In this case, each new column is a new indicator. The new indicators are:

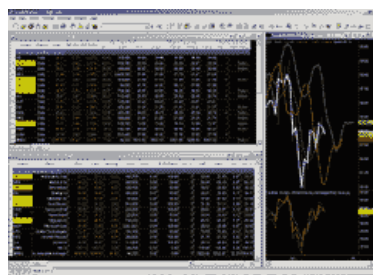


Figure 2: TradeStation, daytrading sample. Here's a sample TradeStation chart demonstrating the high-low strategy. Jacob Singer's spreadsheet can be translated into a RadarScreen.

```

Buy at:
Plot1( DailyOpen + ( High - Low ) / 2 ) ;
Add:
Plot1( ( High - Low ) / 2 );
Gap-Up:
if DailyOpen > High[1] then
Plot1( 1 )
else
Plot1( 0 ) ;
# of Shares:
inputs:
Capital( 10000 ),
Positions( 4 );
Plot1( Round( Capital / Positions / Last / 50,0 ) * 50 ) ;
Performance:
plot1( ( Last - Close[10] ) / Close[10], "Vol" ) ;
Name:
plot1(Description,"Name");

```

The code and workspace will be available for download from the EasyLanguage Exchange at TradeStationWorld at <https://www.tradestationworld.com/discussions/default.asp?Group=8>. Look for the files "Singer - Day Trading.eld" and "Singer.tsw."

--Ian MacAuslan
EasyLanguage Specialist
TradeStation Securities, Inc.
ian@TSSec at EasyLanguage Questions Forum
www.tradestationsupport.com/discussions/forum.asp?FORUM_ID=184

[GO BACK](#)

►METASTOCK: DAYTRADING STRATEGIES

Jacob Singer, in his article "Strategies For Daytrading," presents three systems. Here are MetaStock Explorations to assist you in using these systems.

To use these formulas, go to Tools | The Explorer. Click "New" and then enter the formulas in the respective tabs.

```

Name: High-Low Strategy
COLUMN FORMULAS
Column A: Close
C

```

```

Column B: move
ROC(C,1,$)
Column C: perf.
((C-Ref(C,-10))/Ref(C,-10))*100
Column D: gap up
GapUp()
Column E: high
H
Column F: add
H-L
Filter:
V>500 AND C<10
Name: Pivot-Point Strategy
COLUMN FORMULAS
Column A: buy/sell
C-MP()>0
Column B: perf.
((C-Ref(C,-10))/Ref(C,-10))*100
Column C: 1st stop
(2*MP())-H
Column D: 1st targ
(2*MP())-L
Column E: 2nd stop
MP()-(H-L)
Column F: 2nd targ
MP()+(H-L)
Filter:
V>500 AND C<10 AND colA

```

```

Name: Red-Green Strategy
COLUMN FORMULAS
Column A: Buy
Cross(C,Ref(Mov(H,2,S),-10))
Column B: Sell
Cross(Mov(C,5,S),C)
Column C: Short
Cross(Ref(Mov(L,2,S),-6),C)
Column D: Cover
Cross(C, Mov(C,5,S))

```

--William Golson
Equis Support
www.equis.com

[GO BACK](#)

►AMIBROKER: REVERSE ENGINEERING RSI (II)

In "Reverse Engineering RSI (II)," Giorgos Siligardos discusses new variations of his inverse-RSI indicator initially presented two months ago in the June 2003 STOCKS & COMMODITIES.

The RevEngEMARSI and RevEngTrendRSI indicators can be easily reproduced in AmiBroker using its native AFL language. Listing 1 shows the code that plots the RevEngEMARSI, while Listing 2 shows the code for the RevEngTrendRSI indicator presented in the article.

```

LISTING 1
////////////////////
// RevEngEMARSI
////////////////////
eper = Param("EMA periods", 65, 1, 1000, 1 );
wper = Param("Wilder periods", 14, 1, 100, 1 );
Value = EMA( RSI( wper ), eper );
AUC = Wilders( Max( C - Ref( C, -1 ), 0 ), wper );
ADC = Wilders( Max( Ref( C, -1 ) - C, 0 ), wper );
x = ( wper - 1 )*( ADC * Value/(100-Value)-AUC);
RevEngEMARSI = IIf( x >= 0, C + x, C + x*(100-Value)/Value );
Plot( Close, Date() + " ", Close, colorBlack, styleCandle );
Plot( RevEngEMARSI, "Reverse EMARSI", colorRed );
LISTING 2
////////////////////
// RevEngTrendRSI
////////////////////
RSIperiod = 14;
wper = 14;
Day1 = Param("day for point 1", 12, 1, 31, 1 );
Month1 = Param("month for point 1", 10, 1, 12, 1 );
Year1 = Param("year for point 1", 1990, 1950, 2150, 1 );
Day2 = Param("day for point 2", 9, 1, 31, 1 );
Month2 = Param("month for point 2", 10, 1, 12, 1 );
Year2 = Param("year for point 2", 1992, 1950, 2150, 1 );
Date1 = Day1 == Day() AND Month1 == Month() AND Year1 == Year();
Date2 = Day2 == Day() AND Month2 == Month() AND Year2 == Year();
i = BarsSince( Date1 );
j = BarsSince( Date2 );
RSI1 = ValueWhen( Date1, RSI( RSIperiod ) );
RSI2 = ValueWhen( Date2, RSI( RSIperiod ) );

```

```

Value = Min(Max((RSI2-RSI1)*(i+1)/(i-j)+RSI1, 1 ), 99 );
AUC = Wilders( Max( C - Ref( C, -1 ), 0 ), wper );
ADC = Wilders( Max( Ref( C, -1 ) - C, 0 ), wper );
x = ( wper - 1 )*( ADC * Value/(100-Value)-AUC);
RevEngTrendRSI = If( x >= 0, C + x, C + x*(100-Value)/Value );
Plot( Close, Date() + ", Close", colorBlack, styleCandle );
Plot( RevEngTrendRSI, "Reverse Trend RSI", colorRed );

```

A downloadable version of this formula is available from AmiBroker's website. A sample chart is shown in Figure 3.

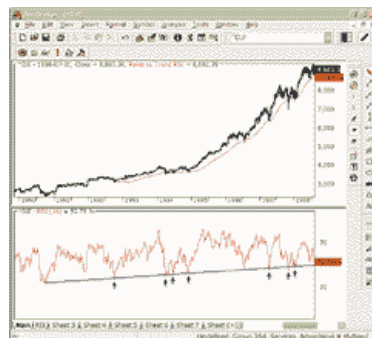


Figure 3: AmiBroker, reverse-engineered RSI. This AmiBroker screenshot shows the weekly Dow Jones Industrials with RevEngTrendRSI (red line in the upper pane) that represents the support and resistance levels of the trendline plotted on the RSI (lower pane).

--Tomasz Janeczko, AmiBroker.com
www.amibroker.com

[GO BACK](#)

▶ AMIBROKER: DAYTRADING STRATEGIES

In "Strategies For Daytrading," Jacob Singer presents three simple trading strategies that can be easily implemented using AmiBroker Formula Language (AFL).

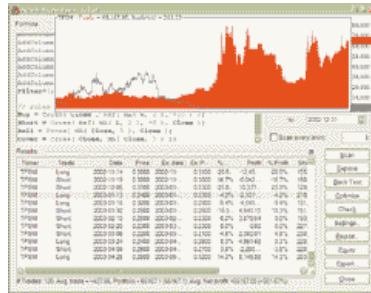
Listing 1 shows the formula for the red-green strategy, and listing 2 shows the formula for high-low strategy. Both formulas can be used in Exploration and Backtest modes. Exploration produces spreadsheet-like output, as described in the article, while backtest mode allows you to evaluate the historical performance of the strategy. We have backtested the red-green strategy on Real Media, Inc. [Tfsm], from September 1998 to May 2003, and it gave more than a 551% return, while a buy-and-hold strategy lost more than 90% during the same period.

```

LISTING 1
////////////////////////////////////
// Red-Green strategy
////////////////////////////////////
// exploration for EOD analysis
AddTextColumn(FullName(),"Name");
AddColumn( C, "Close" );
AddColumn( C - Ref( C, -1 ), "Move" );
AddColumn( V, "Volume", 1 );
AddColumn( ROC( C, 10 ), "Performance");
AddColumn( 2500/C, "# of shares", 1);
AddColumn( GapUp(), "Gap up", 1 );
AddColumn( High, "High");
AddColumn( Null, "Tomorrow's Open");
AddColumn( (H-L)/2, "Add" );
Filter=1;
// rules for backtesting
Buy = Cross( Close , Ref( MA( H, 2 ), -10 ) );
Short = Cross( Ref( MA( L, 2 ), -6 ), Close );
Sell = Cross( MA( Close, 5 ), Close );
Cover = Cross( Close, MA( Close, 5 ) );
LISTING 2
////////////////////////////////////
// High-Low strategy
////////////////////////////////////
// exploration for EOD analysis
AddTextColumn(FullName(),"Name");
AddColumn( C, "Close" );
AddColumn( C - Ref( C, -1 ), "Move" );
AddColumn( V, "Volume", 1 );
AddColumn( ROC( C, 10 ), "Performance");
AddColumn( 2500/C, "# of shares", 1);
AddColumn( GapUp(), "Gap up", 1 );

```

A downloadable version of this formula is available from AmiBroker's website. A sample chart is shown in Figure 4.



--Tomasz Janeczko
www.AmiBroker.com

[GO BACK](#)

AIQ EXPERT DESIGN STUDIO: REVERSE ENGINEERING RSI (II)

```

!!! Stocks & Commodities August 2003 - "Reverse Engineering RSI," by Giorgos Siligardos
!!! Expert Design Studio Code for RevEngSMARSI
!! Set RSI Wilder to 14 in charts.
Close is [close].
Yclose is val([close], 1).
Define ExpPer 27.
Value is simpleavg([rsi wilder], 45).
AUC is expavg(ifff(close > yclose, close - yclose, 0), ExpPer).
ADC is expavg(ifff(yclose > close, yclose - close, 0), ExpPer).
X is (13) * (adc * value / (100 - value) - auc).
RevEngSMARSI is ifff(x>= 0, close + x, close + x * (100 - value)/value).
!!! Stocks & Commodities August 2003 - "Reverse Engineering RSI," by Giorgos Siligardos
!!! Expert Design Studio Code for RevEngEMARSI
!! Set RSI Wilder to 14 in charts.
Close is [close].
Yclose is val([close], 1).
Define ExpPer 27.
Value is expavg([rsi wilder], 45).
AUC is expavg(ifff(close > yclose, close - yclose, 0), ExpPer).
ADC is expavg(ifff(yclose > close, yclose - close, 0), ExpPer).
X is (13) * (adc * value / (100 - value) - auc).
RevEngEMARSI is ifff(x>= 0, close + x, close + x * (100 - value)/value).

```

Sample charts are shown in Figures 5 and 6.

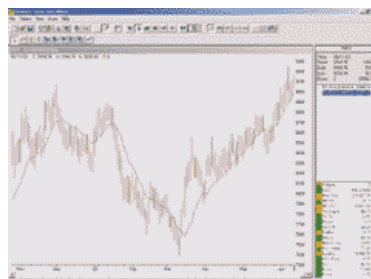


Figure 5: AIQ, reverse-engineered RSI. Here is a sample AIQ chart of the RevEngSMARSI.

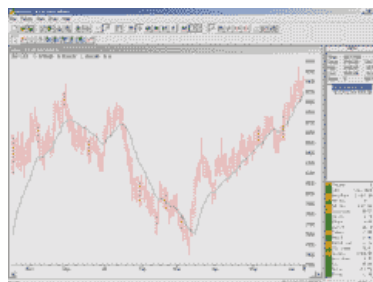


FIGURE 6: AIQ, reverse-engineered RSI. Here is a sample AIQ chart of the RevEngEMARSI.

--Mike Kaden
AIQ Systems
www.aiq.com

[GO BACK](#)

▶ **WEALTH-LAB: REVERSE ENGINEERING RSI (II)**

The RevEngEMARSI and RevEngSMARSI indicators discussed in "Reverse Engineering RSI (II)" by Giorgos Siligardos in this issue return the price level that has to be reached in order for an n-period RSI to touch its p-period exponential or simple moving average (where n and p are two parameters that you can pass to the indicator).

Using different period values for the moving average can yield indicators that have different levels of responsiveness. For example, a lower period moving average will yield an indicator that hugs prices more closely. The simple script given here enters a long position when a shorter-period RevEngEMARSI crosses over a longer-period RevEngEMARSI. A simple profit target and stop loss are used to exit the trades (Figure 7). The code for this system is as follows:



Figure 7: Wealth-Lab, reverse engineering RSI. The Wealth-Lab script given here enters a long position when a shorter-period RevEngEMARSI crosses over a longer-period RevEngEMARSI. A simple profit target and stop loss are used to exit the trades.

```
{ $! 'RevEngEmaRsi' }
var Bar: integer;
var RevEngEmaRsi1, RevEngEmaRsi2: integer;
RevEngEmaRsi1 := RevEngEmaRsiSeries( #Close,65,14 );
PlotSeriesLabel( RevEngEmaRsi1, 0, 505, #Thick, 'RevEngEmaRsi1=RevEngEmaRsi(#Close,65,14)' );
RevEngEmaRsi2 := RevEngEmaRsiSeries( #Close,35,14 );
PlotSeriesLabel( RevEngEmaRsi2, 0, 009, #Thick, 'RevEngEmaRsi2=RevEngEmaRsi(#Close,35,14)' );
InstallProfitTarget( 5 );
InstallStopLoss( 15 );
for Bar := 20 to BarCount - 1 do
begin
if not LastPositionActive then
{ Entry Rules }
begin
if CrossOver( Bar, RevEngEmaRsi2, RevEngEmaRsi1 ) then
BuyAtMarket( Bar + 1, " );
end
else
{ Exit Rules }
begin
ApplyAutoStops( Bar );
end;
end;
end;
```

Both of these indicators are available on the Wealth-Lab.com website and in Wealth-Lab Developer (the desktop application) as pre-made custom indicators.

--Dion Kurczek, Wealth-Lab, Inc.

[GO BACK](#)

WEALTH-LAB: DAYTRADING STRATEGIES

In "Strategies For Daytrading" in this issue, Jacob Singer presents three strategies for daytrading. Each of these strategies can be implemented very easily into Wealth-Lab Developer.

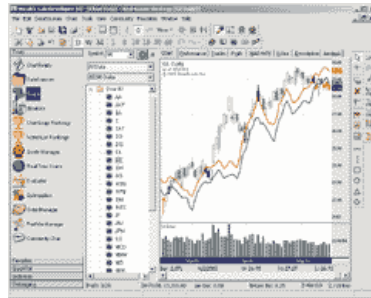


Figure 8: Wealth-Lab, daytrading strategies. The red-green strategy applied to General Electric is displayed in Wealth-Lab. This strategy captured a large positive move in the stock from February to May 2003.

Figure 8 displays a chart of the red-green strategy for General Electric. This strategy captured a large positive move in the stock from February to May 2003. The Wealth-Lab code for the red-green strategy is as follows:

Red-Green Strategy

```
var BAR, H, L: integer;
H := OffsetSeries( SMAseries( #High, 2 ), -10 );
L := OffsetSeries( SMAseries( #Low, 2 ), -10 );
for Bar := 20 to BarCount - 1 do
begin
  if PriceClose( Bar ) > @H[Bar] then
    SetBarColor( Bar, #Green )
  else if PriceClose( Bar ) < @L[Bar] then
    SetBarColor( Bar, #Red );
end;
PlotSeries( H, 0, #Red, #Thick );
PlotSeries( L, 0, #Blue, #Thick );
```

The other two strategies are intended to be run as scans, and they produce a series of output values similar to a spreadsheet. You can use Wealth-Lab Developer's WatchList Scan tool to produce this type of report. You can even sort on any of the individual columns in order to zero in on trading candidates.

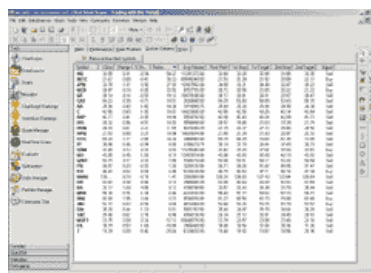


FIGURE 9: Wealth-Lab, daytrading strategies. The trend strategy is displayed, including the first and second stop and target levels.

Figure 9 displays the result of trading with the trend strategy, including the first and second stop and target levels. Here's the Wealth-Lab code for trading with the trend:

```
Trend Strategy
var Bar: integer;
var H, L, TC, C, Change: float;
Bar := BarCount - 1;
C := PriceClose( Bar );
H := PriceHigh( Bar );
L := PriceLow( Bar );
TC := PriceAverageC( Bar );
AddScanColumn( 'Close', C );
Change := C - PriceClose( Bar - 1 );
AddScanColumn( 'Change', Change );
```

```

AddScanColumn( '% Change', Change / PriceClose( Bar - 1 ) * 100 );
AddScanColumn('Performance',100*(C-PriceClose(Bar-104))/PriceClose(Bar-104));
AddScanColumn( 'Avg Volume', SMA( Bar, #Volume, 25 ) );
AddScanColumn( 'Pivot Point', TC );
AddScanColumn( '1st Stop', ( 2 * TC ) - H );
AddScanColumn( '1st Target', ( 2 * TC ) - L );
AddScanColumn( '2nd Stop', TC - ( H - L ) );
AddScanColumn( '2nd Target', TC + ( H - L ) );
if C - TC > 0 then
  AddScanColumnStr( 'Signal', 'Buy' )
else
  AddScanColumnStr( 'Signal', 'Sell' );

```

Figure 10 displays the results of the high-low strategy, sorted by performance. You can update your daily data shortly after the market opens and run this scan to catch today's open price. Based on this information, you can look for viable trades and stocks to keep a closer eye on during the trading session.

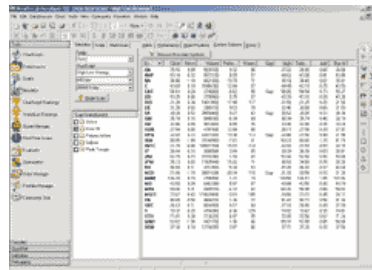


Figure 10: Wealth-Lab, daytrading strategies. The high-low strategy, sorted by performance, is displayed.

```

High-Low Strategy
var Bar, Today: integer;
var C, Add: float;
Bar := BarCount - 2;
Today := BarCount - 1;
C := PriceClose( Bar );
AddScanColumn( 'Close', C );
AddScanColumn( 'Move', C - PriceClose( Bar - 1 ) );
AddScanColumnStr( 'Volume', IntToStr( Trunc( Volume( Bar ) ) ) );
AddScanColumn('Performance',100*(C-PriceClose(Bar-10))/PriceClose(Bar-10));
AddScanColumnStr( 'Shares', IntToStr( Trunc( 2500 / C ) ) );
if PriceLow( Bar ) > PriceHigh( Bar - 1 ) then
  AddScanColumnStr( 'Gap', 'Gap' )
else
  AddScanColumnStr( 'Gap', '' );
AddScanColumn( 'High', PriceHigh( Bar ) );
AddScanColumn( 'Today Open', PriceOpen( Today ) );
Add := ( PriceHigh( Bar ) - PriceLow( Bar ) ) / 2;
AddScanColumn( 'Add', Add );
AddScanColumn( 'Buy At', PriceOpen( Today ) + Add );

```

--Dion Kurczek, *Wealth-Lab, Inc.*
www.wealth-lab.com

[GO BACK](#)

▶NEUROSHELL TRADER: DAYTRADING STRATEGIES

The strategies described in Jacob Singer's article in this issue, "Strategies For Daytrading," are easily recreated in NeuroShell Trader and NeuroShell DayTrader.

To recreate the pivot-point strategy, first create a daily chart, then select "New Trading Strategy ..." from the Insert menu, and enter the following long and short entry and exit conditions in the appropriate locations of the Trading Strategy Wizard:

```

Generate a buy long MARKET order if ALL of the following are true:
And2( A>B(Volume, Mult2(2,Avg(Volume,25)) ),
  A<B(Volume, Mult2(4,Avg(Volume,25)) ) )
A>B ( Close, Avg3(High,Low,Close) )
And2 ( A>B(Lag(Avg3(High,Low,Close),1), Avg3(High,Low,Close)),
  A>B(Lag(Avg3(High,Low,Close),1), Lag(Avg3(High,Low,Close),2))
Generate a long trailing stop order at the following price:
Subtract ( Mult( 2, Avg3(High,Low,Close) ), High )
Generate a sell long MARKET order if ALL of the following are true:
A>B ( Close, Subtract ( Mult( 2, Avg3(High,Low,Close) ), Low ) )
Generate a sell short MARKET order if ALL of the following are true:
And2( A>B(Volume, Mult2(2,Avg(Volume,25)) ),
  A<B(Volume, Mult2(4,Avg(Volume,25)) ) )
A<B ( Close, Avg3(High,Low,Close) )
And2 ( A<B(Lag(Avg3(High,Low,Close),1), Avg3(High,Low,Close)),

```

$A < B(\text{Lag}(\text{Avg3}(\text{High}, \text{Low}, \text{Close}), 1), \text{Lag}(\text{Avg3}(\text{High}, \text{Low}, \text{Close}), 2))$
 Generate a short trailing stop order at the following price:
 $\text{Subtract}(\text{Mult}(2, \text{Avg3}(\text{High}, \text{Low}, \text{Close})), \text{Low})$
 Generate a cover short MARKET order if ALL of the following are true:
 $A < B(\text{Close}, \text{Subtract}(\text{Mult}(2, \text{Avg3}(\text{High}, \text{Low}, \text{Close})), \text{High}))$

To recreate the red-green strategy, create a daily chart, select "New Trading Strategy ..." from the Insert menu and enter the following long and short entry and exit conditions in the appropriate locations of the Trading Strategy Wizard:

Generate a buy long MARKET order if ALL of the following are true:
 $A > B(\text{Close}, \text{Lag}(\text{High}, 10))$
 Generate a sell long MARKET order if ALL of the following are true:
 $\text{Price} < \text{SimpleMovAvg}(\text{Close}, 5)$
 Generate a sell short MARKET order if ALL of the following are true:
 $A < B(\text{Close}, \text{Lag}(\text{Low}, 6))$
 Generate a cover short MARKET order if ALL of the following are true:
 $\text{Price} > \text{SimpleMovAvg}(\text{Close}, 5)$

A sample chart of the red-green strategy is in Figure 11.

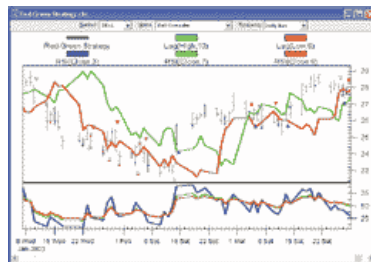


Figure 11: NeuroShell Trader, daytrading strategies. Shown here are sample results of implementing the red-green strategy.

To recreate the high-low strategy, create a five-minute chart (note that you must use the NeuroShell DayTrader to create a five-minute chart), select "New Trading Strategy ..." from the Insert menu and enter the following long and short entry and exit conditions in the appropriate locations of the Trading Strategy Wizard:

Generate a buy long MARKET order if ALL of the following are true:
 $\text{Not}(\text{And2}(A > B(\text{DayOpen}(\text{Date}, \text{Open}, 0), \text{DayHigh}(\text{Date}, \text{High}, 1)), A > B(\text{DayOpen}(\text{Date}, \text{Open}, 1), \text{DayHigh}(\text{Date}, \text{High}, 2))))$
 $A > B(\text{DayHigh}(\text{Date}, \text{High}, 0), \text{Add2}(\text{DayOpen}(\text{Date}, \text{Open}, 0), \text{DayRange}(\text{Date}, \text{High}, \text{Low}, 1)))$
 $A < B(\text{RSI}(\text{Close}, 10), 30)$
 $A > B(\text{Momentum}(\text{RSI}(\text{Close}, 10), 1), 0)$
 Generate a long trailing stop order at the following price:
 $\text{PriceFloor: Percent}(\text{Trading Strategy}, 8)$
 Generate a sell long MARKET order if ALL of the following are true:
 $\text{PriceTarget: Percent}(\text{Trading Strategy}, 8)$
 $\text{And2}(\text{X} \leq \text{Time} \leq \text{Y}(9:30\text{am}, 9:45\text{am}), \text{PriceTarget}(\text{TradingStrategy}, .00001))$
 Generate a sell short MARKET order if ALL of the following are true:
 $\text{Not}(\text{And2}(A < B(\text{DayOpen}(\text{Date}, \text{Open}, 0), \text{DayLow}(\text{Date}, \text{Low}, 1)), A < B(\text{DayOpen}(\text{Date}, \text{Open}, 1), \text{DayLow}(\text{Date}, \text{Low}, 2))))$
 $A < B(\text{DayLow}(\text{Date}, \text{Low}, 0), \text{Subtract}(\text{DayOpen}(\text{Date}, \text{Open}, 0), \text{DayRange}(\text{Date}, \text{High}, \text{Low}, 1)))$
 $A > B(\text{RSI}(\text{Close}, 10), 70)$
 $A < B(\text{Momentum}(\text{RSI}(\text{Close}, 10), 1), 0)$
 Generate a short trailing stop order at the following price:
 $\text{PriceFloor: Percent}(\text{Trading Strategy}, 8)$
 Generate a cover short MARKET order if ALL of the following are true:
 $\text{PriceTarget: Percent}(\text{Trading Strategy}, 8)$
 $\text{And2}(\text{X} \leq \text{Time} \leq \text{Y}(9:30\text{am}, 9:45\text{am}), \text{PriceTarget}(\text{TradingStrategy}, .00001))$

If you have NeuroShell Trader Professional, you can also choose whether the system parameters should be optimized for any of the strategies. After backtesting the strategies, use the "Detailed Analysis ..." button to view the backtest and trade-by-trade statistics for each strategy.

Users of NeuroShell Trader can go to the STOCKS & COMMODITIES section of the NeuroShell Trader free technical support website to download the sample charts, which include the pivot-point strategy, red-green strategy, and the high-low strategy.

--Marge Sherald, Ward Systems Group, Inc.
 301 662-7950, sales@wardsystems.com
www.neuroshell.com

►NEOTICKER: REVERSE ENGINEERING RSI (II)

To implement in NeoTicker the concept presented in "Reverse Engineering RSI (II)" by Giorgos Siligardos in this issue, we will need to create three indicators named RevEngEMARSI (Listing 1), RevEngSMARSI (Listing 2), and RevEngTrendRSI (Listing 3) using the NeoTicker formula language.

The RevEngEMARSI plots the reverse engineered RSI exponential moving average (Figure 12). The RevEngSMARSI plots the reverse engineered RSI simple moving average. The RevEngTrendRSI plots the reverse engineered trend RSI (Figure 13).



FIGURE 12: NeoTicker, reverse-engineered RSI. The RevEngEMARSI plots the reverse-engineered RSI exponential moving average.

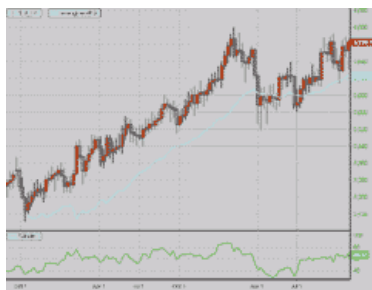


Figure 13: NeoTicker, reverse-engineered RSI. The RevEngSmarsi plots the reverse-engineered RSI simple moving average. The RevEngTrendRSI plots the reverse-engineered trend RSI.

LISTING 1

```
emaper := if (param1 > 1000, 1000, param1);
WildPer := if (param2 > 100, 100, param2);
ExpPer := 2*WildPer-1;
myValue := qc_xaverage(RSIndex(c, WildPer), emaper);
'Average Up Close
myUC := if(c>c(1), c-c(1), 0);
myAUC := qc_xaverage(myUC, ExpPer);
'Average Down Close
myDC := if(c(1)>c, c(1)-c, 0);
myADC := qc_xaverage(myDC, ExpPer);
myx := (WildPer-1)*(myADC*myValue/(100-myValue)-myAUC);
plot1 := if(myx >= 0, c+myx, c+myx*(100-myValue)/myValue);
```

LISTING 2

```
smaper := if(param1> 1000, 1000, param1);
WildPer := if(param2> 100, 100, param2);
ExpPer := 2*WildPer-1;
myValue := average(RSIndex(c, WildPer), smaper-1);
'Average Up Close
myUC := if(c>c(1), c-c(1), 0);
myAUC := qc_xaverage(myUC, ExpPer);
'Average Down Close
myDC := if(c(1)>c, c(1)-c, 0);
myADC := qc_xaverage(myDC, ExpPer);
myx := (WildPer-1)*(myADC*myValue/(100-myValue)-myAUC);
plot1 := if(myx >= 0, c+myx, c+myx*(100-myValue)/myValue);
```

LISTING 3

```
RSIperiod := param1;
day1 := if(param2 > 31, 31, param2);
month1 := if(param3 > 12, 12, param3);
year1 := if(param4 < 1900 or param4 > 3000, 1900, param4);
day2 := if(param5 > 31, 31, param5);
month2 := if(param6 > 12, 12, param6);
year2 := if(param7 < 1900 or param7 > 3000, 1900, param7);
ExpPer := 2*RSIperiod-1;
mydate1 := makedate(year1, month1, day1);
mydate2 := makedate(year2, month2, day2);
BarFromPoint1 := if(data1.dt(0) > mydate1, BarFromPoint1+1, 0);
BarFromPoint2 := if(data1.dt(0) > mydate2, BarFromPoint2+1, 0);
RSI1 := if (BarFromPoint1 > 0, RSIndex(BarFromPoint1-1, c, RSIperiod), 0);
RSI2 := if (BarFromPoint2 > 0, RSIndex(BarFromPoint2-1, c, RSIperiod), 0);
```

```

myValue := Min(Max((RSI2-RSI1)*(BarFromPoint1+1)/
    (BarFromPoint1-BarFromPoint2)+RSI1, 1), 99);
myUC := if(c>c(1), c-c(1), 0);
myAUC := qc_xaverage(myUC, ExpPer);
myDC := if(c(1)>c, c(1)-c, 0);
myADC := qc_xaverage(myDC, ExpPer);
myx := (RSIperiod-1)*(myADC*myValue/(100-myValue)-myAUC);
plot1 := if(myx >= 0, c+myx, c+myx*(100-myValue)/myValue);
success1 := if(data1.dt(0) > mydate2, 1, 0);

```

A downloadable version of the indicators will be available from the NeoTicker Yahoo! User group.

--Kenneth Yuen, TickQuest Inc.
www.tickquest.com

[GO BACK](#)

▶NEOTICKER: DAYTRADING STRATEGIES

To implement in NeoTicker all three trading signals presented in "Strategies For Daytrading" by Jacob Singer, we can use scripting language to build the backtesting systems and apply the quote window sorting functions to pick the right stock.

Pivot-Point Strategy

To implement this strategy, we can use the quote window formula and column sorting functions. Simply create a quote window (Figure 14) similar to the spreadsheet presented in the article. Coloring rules can be applied for each column to create better real-time feedback. The backtesting script ppoint (Listing 1) for this strategy can be constructed using Delphi Script, with trade size as the parameter.



Figure 14: NeoTicker, daytrading strategies. The pivot-point strategy is implemented in NeoTicker. Shown here is a quote window similar to the spreadsheet in Jacob Singer's article.

Red-Green Strategy

The indicator redgreen (Listing 2) presented in the article can be constructed using Delphi Script. This indicator has three parameters for trade size, high lookback period, and low lookback period. This indicator plots the equity, high lookback price, and low lookback price (Figure 15).

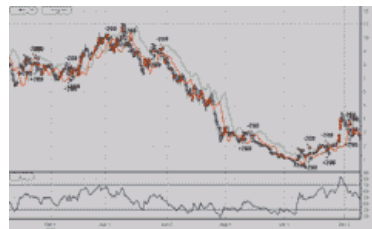


Figure 15: NeoTicker, daytrading strategies. The indicator redgreen presented in Jacob Singer's article plots the equity, high lookback price, and low lookback price.

High-Low Strategy

To implement this strategy, construct a quote window (Figure 16) listing all the potential stocks with their respective buy level.

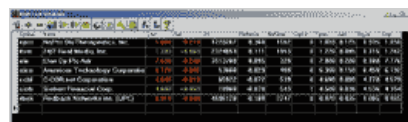


Figure 16: NeoTicker, daytrading strategies. To implement the high-low strategy, construct a quote window listing all the potential stocks with their respective buy levels.

```

LISTING 1
function ppoint : double;
var mycomseries, mycompvol : variant;
    myfilter, bullishsignal, bearishsignal : boolean;
    trueclose, threepp, fivepp : double;
    firststop, firsttarget, secondstop, secondtarget : double;
    mySize : integer;
begin
    mycomseries := itself.CompressSeries ('mys', '1', ppDaily, 1);
    mycompvol := itself.MakeIndicator ('myavol', 'average', ['$mys.v'], ['25']);

```

```

mySize := param1.int;
// Today Pivot Point
trueclose := (data1.close [0] + mycomseries.high [0] +
    mycomseries.low [0])/3;
// Pivot Point three days ago
threepp := (mycomseries.close [2] + mycomseries.high [2] +
    mycomseries.low [2])/3;
// Pivot Point five days ago
fivepp := (mycomseries.close [4] + mycomseries.high [4] +
    mycomseries.low [4])/3;
firststop := 2*trueclose-data1.high [0];
firsttarget := 2*trueclose-data1.low [0];
secondstop := trueclose - (mycomseries.high [0] - mycomseries.low [0]);
secondtarget := trueclose + (mycomseries.high [0] - mycomseries.low [0]);
myfilter := (mycomseries.Volume [0] > 2*mycompvol.Value [1]) and
    (mycomseries.Volume [0] < 4*mycompvol.Value [1]) and
    (NTlib.abs(data1.high [0]-data1.close [0]) < 0.1);
bullishsignal := (data1.close [0] > trueclose) and (trueclose > threepp);
bearishsignal := (data1.close [0] < trueclose) and (trueclose < threepp) and
    (trueclose < fivepp);
if myfilter and bullishsignal then
    trade.longatmarket (mySize, 'Bullish Signal Buy at market');
if myfilter and bearishsignal then
    trade.shortatmarket (mySize, 'Bearish Signal Sell at market');
trade.longexitstop (firststop, Trade.OpenPositionSize, otfFillorKill,
    'long stop exist with first stop');
trade.longexitlimit (firsttarget, Trade.OpenPositionSize, otfFillorKill,
    'long target exist with first target');
trade.longexitstop (secondstop, Trade.OpenPositionSize, otfFillorKill,
    'long stop exist with second stop');
trade.longexitlimit (secondtarget, Trade.OpenPositionSize, otfFillorKill,
    'long target exist with second target');
trade.shortexitstop (firststop, Trade.OpenPositionSize, otfFillorKill,
    'short stop exist with first stop');
trade.shortexitlimit (firsttarget, Trade.OpenPositionSize, otfFillorKill,
    'short target exist with first target');
trade.shortexitstop (secondstop, Trade.OpenPositionSize, otfFillorKill,
    'short stop exist with second stop');
trade.shortexitlimit (secondtarget, Trade.OpenPositionSize, otfFillorKill,
    'short target exist with second target');
result := trade.CurrentEquity;
end;
LISTING 2
function redgreen : double;
var mycomseries, dailyhigh, dailylow : variant;
    bullishsignal, bearishsignal : boolean;
    mySize, highPeriod, lowPeriod : integer;
begin
    mycomseries := itself.CompressSeries ('mys', '1', ppDaily, 1);
    dailyhigh := itself.Makeindicator ('dhigh', 'average', ['$mys.h'], ['2']);
    dailylow := itself.Makeindicator ('dlow', 'average', ['$mys.l'], ['2']);
    mySize := param1.int;
    highPeriod := param2.int-1;
    lowPeriod := param3.int-1;
    bullishsignal := (dailyhigh.Value [highPeriod] < data1.close [0]) and
        (data1.barsnum [0] > highPeriod);
    bearishsignal := (dailylow.value [lowPeriod] > data1.close [0]) and
        (data1.barsnum [0] > lowPeriod);
    if bullishsignal then
        trade.longatmarket (mySize, 'Long close gt high of 10 days ago');
    if bearishsignal then
        trade.shortatmarket (mySize, 'Short close lt low of 6 days ago');
    itself.plot [1] := trade.currentequity;
    if (data1.barsnum [0] > highPeriod) and (data1.barsnum [0] > lowPeriod) then
        begin
            itself.plot [2] := dailyhigh.Value [highPeriod];
            itself.plot [3] := dailylow.Value [lowPeriod];
        end
    else
        begin
            itself.successex [2] := false;
            itself.successex [3] := false;
        end;
    end;
end;

```

These quote windows and backtesting systems will be available for download from the Yahoo! NeoTicker user group file area at <http://groups.yahoo.com/group/neoticker/>.

--Kenneth Yuen, TickQuest Inc.
www.tickquest.com

[GO BACK](#)

In his article "Reverse Engineering RSI (II)," Giorgos Siligardos presents additional functions for use with his reverse-engineered RSI.

The three new functions share several common characteristics that can be represented as subfunctions.

```
Name: Reverse Engineered RSI AUC
Short Name: RevEngRSIAUC
Inputs: Close, ExpPer
Formula:
EMA (If (Inc (Close),Change (Close,1),0),ExpPer)
Name: Reverse Engineered RSI ADC
Short Name: RevEngRSIADC
Inputs: Close, ExpPer
Formula:
EMA (If (Dec (Close),Negate (Change (Close,1)),0),ExpPer)
Name: Reverse Engineered RSI Value Delta
Short Name: RevEngRSIValueDelta
Inputs: Subvalue, Close, Wilder Periods, ExpPer
Formula:
Mult (Sub (Wilder Periods,1),Sub (Mult (RevEngRSIADC (Close,ExpPer),Div
(Subvalue,Sub (100, Subvalue))),RevEngRSIAUC (Close,ExpPer)))
Name: Reverse Engineered RSI Value
Short Name: RevEngRSIValue
Inputs: Subvalue, Close, Wilder Periods, ExpPer
Formula:
If (GE (RevEngRSIValueDelta (Subvalue,Close,Wilder Periods,ExpPer),0),Add
(Close, RevEngRSIValueDelta (Subvalue,Close,Wilder Periods,ExpPer)),Add
(Close,Mult (RevEngRSIValueDelta (Subvalue,Close,Wilder Periods,ExpPer),Div
(Sub (100, Subvalue), Subvalue))))
```

Using these subfunctions, the new reverse-engineered functions can be written as follows. Note that the value of ExpPer should be calculated externally and is equal to 2*Wilder-1.

```
Name: Reverse Engineered EMA RSI
Short Name: RevEngEMARSI
Inputs: Close, EMA Time Periods, Wilder Time Periods, ExpPer
Formula:
RevEngRSIValue (EMA (RSI (Close,Wilder Time Periods),
EMA Time Periods),Close,Wilder Time Periods, ExpPer)
Name: Reverse Engineered SMA RSI
Short Name: RevEngSMARSI
Inputs: Close, SMA Time Periods Minus One, Wilder Time Periods, ExpPer
Formula:
RevEngRSIValue (MA (RSI (Close,Wilder Time Periods),
SMA Time Periods Minus One),Close,Wilder Time Periods,ExpPer)
Name: Reverse Engineered Trend RSI
Short Name: RevEngTrendRSI
Inputs: Close, RSI Period, ExpPer, Month1, Day1, Year1, Month2, Day2, Year2
Formula:
RevEngRSIValue (Min (Max (Add (Mult (Sub (ValueOn (RSI (Close,
RSI Period),Year2,Month2,Day2), ValueOn (RSI (Close,RSI Period),
Year1,Month1,Day1)),Div (Add (BarsSinceDate (Year1,Month1,Day1),
1),Sub (BarsSinceDate (Year1,Month1,Day1),BarsSinceDate
(Year2,Month2,Day2))))),ValueOn (RSI (Close,RSI Period),Year1,
Month1,Day1)),1),99),Close,RSI Period,ExpPer)
```

Note that the value of the trend RSI will not be valid prior to date 2.

These functions are available in a function file that can be downloaded from the TradingSolutions website in the Solution Library section.

--Gary Geniesse
NeuroDimension, Inc.
800 634-3327, 352 377-5144
www.tradingsolutions.com

[GO BACK](#)

▶INVESTOR/RT: DAYTRADING STRATEGIES

In "Strategies For Daytrading," Jacob Singer discusses the use of RSI and moving average bands in five-, three-, and 60-minute charts. The chart in Figure 17 demonstrates the ability of Investor/RT to represent the data of all three time frames, along with indicators computed on each individual time frame, within one multiperiodicity chart.

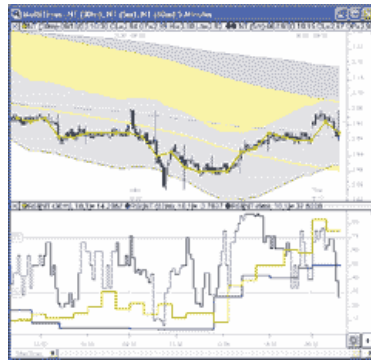


Figure 17: Investor/RT, multiperiodicity chart. In this Investor/RT chart, the top pane shows Nortel Networks (NT) as five-minute candles (black), a 30-minute connected line (gold), and 60-minute bars (blue). Moving average bands are also drawn for each time frame (using the price bands indicator) in their corresponding colors (5=gray, 30=gold, 60=blue). In the lower pane, the RSI is drawn for each periodicity, also in matching colors.

The price bands are drawn using the preferences seen in Figure 18.

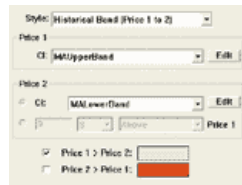


Figure 18: Preferences for the moving average bands on five-minute candles. Historical bands are drawn between the two custom indicators "MAUpperBand" ($MA * 1.03$) and "MALLowerBand" ($MA * 0.97$) using these preferences.

The historical price bands are drawn between two custom indicators. The custom indicator MAUpperBand represents a line 3% above the 28-period moving average ($MA * 1.03$). The custom indicator MALLowerBand represents a line 3% below the 28-period moving average ($MA * 0.97$). The price band indicator allows the user to draw a solid band such as the three seen in Figure 17 between any two custom indicator lines. This same price band indicator is also added to the 30-minute and 60-minute data, with only the color changing (gold=30, blue=60).

The lower pane of the chart in Figure 17 shows the 10-period Rsi for all three time frames, drawn as stepped lines. The preferences for the RSI drawn on the five-minute data can be seen in Figure 19.

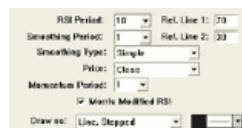


Figure 19: RSI preferences. The lower pane in Figure 17 shows the 10-period RSI for all three time frames, drawn as stepped lines. The preferences for the RSI drawn on the five-minute data can be seen here.

By overlaying the RSIs of all three time frames within one pane, the user can quickly see when all three have simultaneously exceeded a certain threshold value (20, 30, 70, 80, etc.). Any indicator can be overlaid in this multiperiodicity environment, using a mixture of any number of periodicities (one-minute, 39-minute, five-second, 22-tickbar, etc.).

For more information, see:

Multiperiodicity charts: <http://www.linnsoft.com/tour/multiPerCharts.htm>
 Price bands indicator: <http://www.linnsoft.com/tour/techind/bands.htm>
 Custom indicators: <http://www.linnsoft.com/tour/customIndicator.htm>
 Relative strength index (RSI): <http://www.linnsoft.com/tour/techind/rsi.htm>

--Chad Payne, Linn Software
 800 546-6842, info@linnsoft.com
www.linnsoft.com

[GO BACK](#)

Here are the StockWiz formulas for the strategies described in "Strategies For Daytrading" by Jacob Singer in this issue.

```
#####
# STOCKWIZ: PIVOT-POINT STRATEGY
# This StockWiz formula ranks all companies for trading based on the true close
# or pivot point strategy that is presented in Jacob Singer's article
# "Strategies For # Daytrading" in the August 2003 issue of the
# Stocks & Commodities magazine.
#####
# Clear all contents of output grid
(CLEAR)
(SOURCE "WORKING_GROUP")
# Set number of rows in worksheet to the number of entries in the Working Group.
# Variable I keeps track of the number of companies processed so far.
# Variable J keeps track of the number of companies added to the worksheet.
(SET I 0)
(SET J 0)
(SET TOTAL (DBSIZE))
# Set labels in grid
(GRIDFIELD "Ticker" "STRING" "10")
(GRIDFIELD "Name" "STRING" "25")
(GRIDFIELD "Close" "STRING" "7")
(GRIDFIELD "Change" "STRING" "7")
(GRIDFIELD "PercChange" "STRING" "12")
(GRIDFIELD "Performance" "STRING" "12")
(GRIDFIELD "Volume" "STRING" "10")
(GRIDFIELD "AvgVol25days" "STRING" "12")
(GRIDFIELD "PivotPoint" "STRING" "10")
(GRIDFIELD "Stop1" "STRING" "7")
(GRIDFIELD "Target1" "STRING" "7")
(GRIDFIELD "Stop2" "STRING" "7")
(GRIDFIELD "Target2" "STRING" "7")
(GRIDFIELD "BUYorSELL" "STRING" "10")
# Load the first company from the list in the Working Group
(SET STATUS (LOADFIRST))
(GOTO %ERROR (NE STATUS 0))
%NEXT:
(SET HIGH (GETVECTOR (CURRENT) "HIGH"))
(SET LOW (GETVECTOR (CURRENT) "LOW"))
(SET CLOSE (GETVECTOR (CURRENT) "CLOSE"))
(SET VOLUME (GETVECTOR (CURRENT) "VOLUME"))
(SET I (ADD I 1))
(SET LASTDATE (LASTDATE))
# Skip over any companies with less than 104 prices
(GOTO %UPDATE (LE (VSIZE CLOSE) 104))
# Skip over if this company has bad data - such as lot's of missing values, etc.
(GOTO %UPDATE (BADDATA CLOSE LASTDATE))
(SET LASTCLOSE (GETDOUBLE "LastClose"))
(SET C (VSIZE CLOSE))
(SET C (SUB C 2))
(SET CLOSE-2 (VGET CLOSE C))
(SET CHANGE (SUB LASTCLOSE CLOSE-2)) # Change: today's minus yesterday's close
(SET PERCHG (GETDOUBLE "PerChgFromPrevDay")) # Percentage change since yesterday's Close
(SET MOM (MOMENTUM CLOSE 104))
(SET PERFORMANCE (VSSUB MOM 100)) # 104 day ROC
(SET P (VSIZE PERFORMANCE))
(SET P (SUB P 1))
(SET PERF (VGET PERFORMANCE P)) # Last Performance value
(SET LASTVOLUME (GETDOUBLE "LastVolume"))
(SET FIRSTDATE (DATEBSUB LASTDATE 25))
(SET VVOLUME (VEC2VEC VOLUME FIRSTDATE LASTDATE))
(SET AVGVOL (MEAN VVOLUME)) # Average Volume over the last 25 days
(SET HL (VADD HIGH LOW))
(SET HLC (VADD HL CLOSE))
(SET PIVOT (VSDIV HLC 3)) # Pivot Point - true close
(SET V (VSIZE PIVOT))
(SET V (SUB V 1))
(SET LASTPIVOT (VGET PIVOT V)) # Last Pivot value
(SET LASTHIGH (GETDOUBLE "LastHigh"))
(SET LASTLOW (GETDOUBLE "LastLow"))
(SET ST1 (MUL LASTPIVOT 2))
(SET STOP1 (SUB ST1 LASTHIGH))
(SET TGT1 (MUL LASTPIVOT 2))
(SET TARGET1 (SUB TGT1 LASTLOW))
(SET ND2 (SUB LASTHIGH LASTLOW))
(SET STOP2 (SUB LASTPIVOT ND2))
(SET TGT2 (SUB LASTHIGH LASTLOW))
(SET TARGET2 (ADD LASTPIVOT TGT2))
(SET C-P (SUB LASTCLOSE LASTPIVOT))
(GOTO %BUY (GT C-P 0)) # Buy if Close minus Pivot is positive
(GOTO %SELL (LT C-P 0)) # Sell if Close minus Pivot is negative
%BUY: (SET B-S 1) # Set 1 for Buy
(GOTO %SKIP (GT C-P 0))
%SELL: (SET B-S 0) # Set 0 for Sell
%SKIP:
(SET J (ADD J 1))
(SET TICKER (CURRENT))
(GRID TICKER "Name" (GETSTRING "NAME"))
```

```

(GRID TICKER "Close" (DOUBL2STR LASTCLOSE "%.2lf"))
(GRID TICKER "Change" (DOUBL2STR CHANGE "%.2lf"))
(GRID TICKER "PercChange" (DOUBL2STR PERCHG "%.2lf"))
(GRID TICKER "Performance" (DOUBL2STR PERF "%.2lf"))
(GRID TICKER "Volume" (DOUBL2STR LASTVOLUME "%.0lf"))
(GRID TICKER "AvgVol25days" (DOUBL2STR AVGVOL "%.0lf"))
(GRID TICKER "PivotPoint" (DOUBL2STR LASTPIVOT "%.2lf"))
(GRID TICKER "Stop1" (DOUBL2STR STOP1 "%.2lf"))
(GRID TICKER "Target1" (DOUBL2STR TARGET1 "%.2lf"))
(GRID TICKER "Stop2" (DOUBL2STR STOP2 "%.2lf"))
(GRID TICKER "Target2" (DOUBL2STR TARGET2 "%.2lf"))
(GRID TICKER "BUYorSELL" (DOUBL2STR B-S "%.0lf"))
%UPDATE:
  (SET I (ADD I 1))
  (PROGRESS 0 TOTAL I)
  (SET STATUS (LOADNEXT))
  (GOTO %NEXT (EQ STATUS 0))
# Terminate program if user clicked on the 'Cancel' button
  (GOTO %EXIT (ESCAPE))
# Sort all companies by Performance
%SORT: (GRIDSORT "Performance" "DESCENDING")
  (GOTO %EXIT (TRUE))
%ERROR: (MESSAGE "An error has occurred - Unable to continue")
%EXIT: (EXIT STATUS)
-----

#####
STOCKWIZ: RED-GREEN STRATEGY
# This StockWiz formula issues trading signals based on previous highs and
# lows reached by the shares. The Red-Green strategy is presented in Jacob
# Singer's article "Strategies For Daytrading" in the August 2003 issue of
# Stocks & Commodities magazine.
#####
# Clear all contents of the worksheet
  (CLEAR)
  (SOURCE "WORKING_GROUP")
# Set number of rows in worksheet to the number of entries in the Working Group.
# Variable I keeps track of the number of companies processed so far.
  (SET I 0)
  (SET TOTAL (DBSIZE))
# Write labels to the output grid
  (GRIDFIELD "Ticker" "STRING" "10")
  (GRIDFIELD "Name" "STRING" "25")
  (GRIDFIELD "LastClose" "STRING" "10")
  (GRIDFIELD "Signal" "STRING" "10")
# Load the first company from the list in the Working Group
  (SETSTATUS (LOADFIRST))
  (GOTO %ERROR (NE STATUS 0))
%NEXT:(SET CLOSE (GETVECTOR (CURRENT) "CLOSE"))
  (SET HIGH(GETVECTOR (CURRENT) "HIGH"))
  (SET LOW(GETVECTOR (CURRENT) "LOW"))
  (SET I (ADD I 1))
# Skip over this company if all values are NA
  (SET SIGNAL "")
  (GOTO %UPDATE (ISVNA CLOSE))
  (SET LASTCLOSE (GETDOUBLE "LastClose"))
  (SET H (VSIZE HIGH))
  (SET H (SUB H 10))
  (SET H-10 (VGET HIGH H)) # Get the high 10 days ago
  (SET L (VSIZE LOW))
  (SET L (SUB L 6))
  (SET L-6 (VGET LOW L)) # Get the low 6 days ago
  (SET MA (MOVAVG CLOSE 5)) # 5 day moving average [MA] of the close
  (SET M (VSIZE MA))
  (SET M (SUB M 1))
  (SET LAST_MA (VGET MA M)) # Get the last MA value
  (GOTO %NEXT1 (GT LASTCLOSE H-10)) # Go to BUY if the LastClose exceeds the High 10 days ago
%NEXT1:(GOTO %BUY(GT LASTCLOSE LAST_MA)) # Go to BUY if the latest price available crosses above its 5 day MA
  (GOTO %NEXT2 (LT LASTCLOSE L-6)) # Go to SELL if the LastClose is less than the Low 6 days ago
%NEXT2:(GOTO %SELL(LE LASTCLOSE LAST_MA)) # Go to SELL if the latest price available crosses below its 5 day MA
%BUY:(SET SIGNAL "BUY")
  (GRID(CURRENT) "Name" (GETSTRING "NAME"))
  (GRID(CURRENT) "LastClose" (DOUBL2STR LASTCLOSE "%.2lf"))
  (GRID(CURRENT) "Signal" SIGNAL)
  (GOTO %UPDATE (TRUE))
%SELL:(SET SIGNAL "SELL")
  (GRID(CURRENT) "Name" (GETSTRING "NAME"))
  (GRID(CURRENT) "LastClose" (DOUBL2STR LASTCLOSE "%.2lf"))
  (GRID(CURRENT) "Signal" SIGNAL)
  (GOTO %UPDATE (TRUE))
%UPDATE:
# Display the number of records processed in the status bar
  (STATUSBAR 1 (DOUBL2STR I "%.0lf"))
  (PROGRESS 0 TOTAL I)
# Terminate the program if the user clicked on the 'Cancel' button
  (GOTO %EXIT (ESCAPE))
  (SETSTATUS (LOADNEXT))
  (GOTO %NEXT (EQ STATUS 0))
  (GOTO %EXIT (TRUE))
%ERROR:(MESSAGE "An error has occurred - Unable to continue")

```

%EXIT:(EXIT 0)

#####

STOCKWIZ: HIGH-LOW STRATEGY

This StockWiz formula screens all companies for trading based

on the High-Low strategy that is presented in Jacob Singer's article

"Strategies For Daytrading" in the August 2003 issue of the Stocks &

Commodities magazine. The value for the "NextOpen" column [SET

OPEN_t+1] must be filled in manually with tomorrow's Open price.

#####

Clear all contents of the worksheet

(CLEAR)

(SOURCE "WORKING_GROUP")

Set number of rows in worksheet to the number of entries in the Working Group.

Variable I keeps track of the number of companies processed so far.

(SET I 0)

(SET TOTAL (DBSIZE))

Write labels to the output grid

(GRIDFIELD "Ticker" "STRING" "10")

(GRIDFIELD "Name" "STRING" "25")

(GRIDFIELD "LastClose" "STRING" "10")

(GRIDFIELD "Move" "STRING" "7")

(GRIDFIELD "Volume" "STRING" "10")

(GRIDFIELD "Performance" "STRING" "12")

(GRIDFIELD "NoShares" "STRING" "10")

(GRIDFIELD "GapUp" "STRING" "7")

(GRIDFIELD "High" "STRING" "7")

(GRIDFIELD "NextOpen" "STRING" "10")

(GRIDFIELD "Add" "STRING" "7")

(GRIDFIELD "BuyAt" "STRING" "7")

(GRIDFIELD "Stop" "STRING" "7")

Load the first company from the list in the Working Group

(SETSTATUS (LOADFIRST))

(GOTO %ERROR (NE STATUS 0))

%AGAIN:

(SET I (ADD I 1))

Calculate the columns for the worksheet

(SET LASTCLOSE (GETDOUBLE "LastClose")) # Get the last close price

(SET CLOSE(GETVECTOR (CURRENT) "CLOSE"))

(SET C (VSIZE CLOSE))

(SET C (SUB C 2))

(SET C-Yest (VGET CLOSE C))

(SET MOVE (SUB LASTCLOSE C-Yest)) # Price move since yesterday

(SET LASTVOLUME (GETDOUBLE "LastVolume")) # Last Volume

(SET MOM (MOMENTUM CLOSE 10))

(SET PERFORM (VSSUB MOM 100)) # 10 day ROC

(SET P (VSIZE PERFORM))

(SET P (SUB P 1))

(SET PERF (VGET PERFORM P)) # Last Performance value

(SET CAPITAL (DIV 10000 4))

(SET SHARES2 (DIV CAPITAL LASTCLOSE))

(SET SHARES(INTEGER SHARES2)) # Number of shares to purchase in each trade

(SET LASTLOW (GETDOUBLE "LastLow"))

(SET HIGH(GETVECTOR (CURRENT) "HIGH"))

(SET H (VSIZE HIGH))

(SET H (SUB H 2))

(SET H-Yest (VGET HIGH H)) # Yesterdays' High price

(GOTO %GAPUP1 (GT LASTLOWH-Yest))

(GOTO %GAPUP0 (LE LASTLOWH-Yest))

%GAPUP1: (SET GAP 1) # Set 1 if there is a previous gap

(GOTO %SKIP (GT LASTLOW H-Yest)) # Skip GAPUP0

%GAPUP0: (SET GAP 0) # Set 0 if there is not a previous gap

%SKIP:

(SET LASTHIGH (GETDOUBLE "LastHigh")) # Get the last High

(SET OPEN_t+1 0) # Tomorrows open, filled in manually

(SET ADD1 (SUB LASTHIGH LASTLOW))

(SET ADD-TR (DIV ADD1 2)) # The 'Add' column: [High-Low]/2

(SET BUY_AT (ADD OPEN_t+1 ADD-TR)) # Tomorrows open plus 'Add'

(SET STOP1 (MUL BUY_AT 0.04))

(SET STOP(SUB BUY_AT STOP1)) # Stop-loss level

(SET TICKER (CURRENT))

(GRID TICKER "Name" (GETSTRING "NAME"))

(GRID TICKER "LastClose" (DOUBL2STR LASTCLOSE "%.2lf"))

(GRID TICKER "Move" (DOUBL2STR MOVE "%.2lf"))

(GRID TICKER "Volume" (DOUBL2STR LASTVOLUME "%.0lf"))

(GRID TICKER "Performance" (DOUBL2STR PERF "%.2lf"))

(GRID TICKER "NoShares" (DOUBL2STR SHARES "%.0lf"))

(GRID TICKER "GapUp" (DOUBL2STR GAP "%.0lf"))

(GRID TICKER "High" (DOUBL2STR LASTHIGH "%.2lf"))

(GRID TICKER "NextOpen" (DOUBL2STR OPEN_t+1 "%.2lf"))

(GRID TICKER "Add" (DOUBL2STR ADD-TR "%.2lf"))

(GRID TICKER "BuyAt" (DOUBL2STR BUY_AT "%.2lf"))

(GRID TICKER "Stop" (DOUBL2STR STOP "%.2lf"))

NOTE:The values in the "BuyAt" and "Stop" columns will change automatically

after we manually fill the "NextOpen" column with tomorrows Open prices for

each company

```

(SET STATUS (LOADNEXT))
(GOTO %AGAIN (EQ STATUS 0))
# Sort all companies by Performance
(GRIDSORT "Performance" "DESCENDING")
# Terminate program if user clicked on the 'Cancel' button
(GOTO %EXIT (ESCAPE))
%ERROR:(MESSAGE "An error has occurred - Unable to continue")
%EXIT:(EXIT STATUS)

```

--Steve Kalhas
StockWiz.com

[GO BACK](#)

▶ ASPEN RESEARCH: REVERSE ENGINEERING RSI (II)

In "Reverse Engineering RSI (II)," author Giorgos Siligardos continues his work from the June 2003 S&C and presents several more indicators based on RSI that are designed to simplify price projection and get a clearer visual representation with these three indicators.

The screenshots in Figures 20 and 21 show how the Ema_Rsi and Sma_Rsi studies would look on an Aspen Graphics chart. The Ema_Rsi study is drawn with a green line along with an EmaPer value of 21. The Sma_Rsi study is drawn with a cyan line along with a SmaPer value of 21.



Figure 20: Aspen Graphics, Ema_Rsi. The Ema_Rsi study is drawn with a green line along with an EmaPer value of 21.

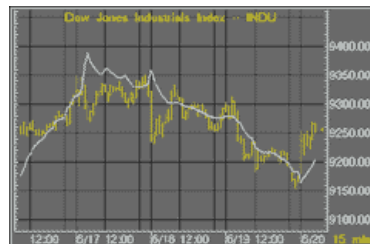


Figure 21: Aspen Graphics, Sma_Rsi. The Sma_Rsi study is being drawn with a cyan line along with a SmaPer value of 21.

The Ema_Rsi is a study that can be overlaid onto any chart. Once it is placed onto the chart, the EmaPer value will need to be set so that the formula will calculate the exponential moving average of the Rsi.

```

EMA_RSI(input,EmaPer=1,Periods=14)={
  retval = 0
  ExpPer = 2 * Periods - 1
  value = eavg(rsi($1,periods),EmaPer)
  auc = eavg(if($1.close>$1.close[1],$1.close-$1.close[1],0),(2 * periods - 1))
  adc = eavg(if($1.close<$1.close[1],$1.close[1]-$1.close,0),(2 * periods - 1))
  x = (Periods - 1) * (adc * value / (100 - value) - auc)
  if x >= 0 then retval = $1.close + x else retval = $1.close + x * ((100 - value) / value)
  retval
}

```

The Sma_RSI is a study that also can be overlaid onto any chart. Once it is placed onto the chart, the SmaPer value will need to be set so that the formula will calculate the simple moving average of the RSI.

```

SMA_RSI(input,SmaPer=1,Periods=14)={
  retval = 0
  ExpPer = 2 * Periods - 1
  value = savg(rsi($1,periods),SmaPer - 1)
  auc = eavg(if($1.close>$1.close[1],$1.close-$1.close[1],0),(2 * periods - 1))

```

```

adc = eavg(if($1.close<$1.close[1],$1.close[1]-$1.close,0),(2 * periods - 1))
x = (Periods - 1) * (adc * value / (100 - value) - auc)
if x >= 0 then retval = $1.close + x else retval = $1.close + x * ((100 - value) / value)
retval
}

```

These studies were written to work with the DTN datafeed. Studies can be made available for the Reuters feed upon request.

--Andy Sewell
 Aspen Graphics Technical Support
 970 945-2921
 ASewell@aspenres.com
 www.aspenres.com

[GO BACK](#)

▶TECHNIFILTER PLUS: REVERSE ENGINEERING RSI (II)

Here are the RevEngEMARSI and RevEngSMARSI formulas discussed by Giorgos Siligardos in "Reverse Engineering Rsi (II)."

The formulas take two parameters: The first is the RSI period, and the second is the average period value required by each formula.

Reverse Engineering Exponential Average of RSI
 NAME: RevEngEMARSI
 SWITCHES: multiline
 PARAMETERS: 14,45
 FORMULA:
 [1]: 2 * &1 - 1 {WildPer}
 [2]: CG&1X&2 {value}
 [3]: (C-CY1)U4X[1] {AUC}
 [4]: (CY1-C)U4X[1] {ADC}
 [5]: ([1]-1) * ([4] * [2]/(100-[2]) - [3]) {x}
 [6]: ([5] > 0) * (C + [5]) + (T=0) * (C + [5] * (100-[2])/[2])
 Reverse Engineering Simple Average of RSI
 NAME: RevEngSMARSI
 SWITCHES: multiline
 PARAMETERS: 14,45
 FORMULA:
 [1]: 2 * &1 - 1 {WildPer}
 [2]: &2 - 1 {SMAPer-1}
 [3]: CG&1A[2] {value}
 [4]: (C-CY1)U4X[1] {AUC}
 [5]: (CY1-C)U4X[1] {ADC}
 [6]: ([1]-1) * ([5] * [3]/(100-[3]) - [4]) {x}
 [7]: ([6] > 0) * (C + [6]) + (T=0) * (C + [6] * (100-[3])/[3])

Visit RTR's website at <http://www.rtrsoftware.com> to download these formulas as well as program updates.

--Clay Burch, RTR Software
 919 510-0608, rtrsoft@aol.com
 www.rtrsoftware.com

[GO BACK](#)

All rights reserved. © Copyright 2003, Technical Analysis, Inc.

Return to [August 2003 Contents](#)
