January 2015

For this month's Traders' Tips, the focus is John Ehlers' article in this issue, "Whiter Is Brighter." Here, we present the January 2015 Traders' Tips code with possible implementations in various software.

Code for TradeStation is already provided in Ehlers' article by the author. S&C subscribers will find that code at the Subscriber Area of our website here. Presented here is an overview of some possible implementations for other software as well.

Traders' Tips code is provided to help the reader implement a selected technique from an article in this issue or another recent issue. The entries here are contributed by various software developers or programmers for software that is capable of customization.

TRADESTATION: JANUARY 2015

In "Whiter Is Brighter," author John Ehlers presents a new indicator he calls the *universal oscillator*. It is based on his theory that market data resembles pink noise, or as he puts it, "noise with memory."

In his article, Ehlers has already provided some TradeStation code for his ultimate oscillator that could be used to create a short-term trading strategy. To facilitate using it with the TradeStation Scanner, we have added alerts to the indicator and have provided an example short-term trading strategy. You can apply the strategy to a TradeStation chart to perform a backtest on a symbol of

your choice or you can use the strategy along with the TradeStation Portfolio Maestro to backtest your favorite portfolio.

For convenience, we are providing this indicator code with alerts as well as the example strategy at our TradeStation and EasyLanguage support forum at http://www.tradestation.com/TASC-2015. The ELD filename is "_TASC_UniversalOscillator.ELD." For more information about EasyLanguage in general, please see http://www.tradestation.com/EL-FAQ. The code is also shown below:

```
_Ehlers_Universal Oscillator (Indicator)

// Universal Oscillator
// (c) 2014 John F. Ehlers
// TASC January 2015
inputs:
        BandEdge( 20 ) ;
variables:
        WhiteNoise( 0 ),
        a1( 0 ),
        b1( 0 ),
        c1( 0 ),
        c2( 0 ),
        c3( 0 ),
        Filt(0),
        Peak(0),
        Universal( 0 ) ;

once
        begin
        if BandEdge <= 0 then
                RaiseRunTimeError( "BandEdge must be > zero" ) ;
        end ;

WhiteNoise = ( Close - Close[2] ) / 2 ;

// SuperSmoother Filter
a1 = ExpValue( -1.414 * 3.14159 / BandEdge ) ;
b1 = 2 * a1 * Cosine( 1.414 * 180 / BandEdge ) ;
c2 = b1 ;
c3 = -a1 * a1 ;
c1 = 1 - c2 - c3 ;
Filt = c1 * ( WhiteNoise + WhiteNoise [1] ) / 2 +
c2 * Filt[1] + c3 * Filt[2] ;
If Currentbar = 1 then
        Filt = 0 ;
If Currentbar = 2 then
        Filt = c1 * 0 * ( Close + Close[1] ) / 2 + c2 * Filt[1] ;
If Currentbar = 3 then
        Filt = c1 * 0 * ( Close + Close[1] ) / 2 + c2 * Filt[1] +
c3 * Filt[2] ;

// Automatic Gain Control (AGC)
Peak = .991 * Peak[1] ;
If Currentbar = 1 then
        Peak = .0000001 ;
If AbsValue( Filt ) > Peak then
        Peak = AbsValue( Filt ) ;
```

```
If Peak <> 0 then
        Universal = Filt / Peak ;
Plot1( Universal ) ;
Plot2( 0 ) ;


if Universal crosses over 0 then
        Alert( "Osc cross over zero line" )
else if Universal crosses under 0 then
        Alert( "Osc cross under zero line" ) ;


_Ehlers_Universal Oscillator (Strategy)

// Universal Oscillator
// (c) 2014 John F. Ehlers
// TASC January 2015
inputs:
        BandEdge( 20 ) ;
variables:
        WhiteNoise( 0 ),
        a1( 0 ),
        b1( 0 ),
        c1( 0 ),
        c2( 0 ),
        c3( 0 ),
        Filt(0),
        Peak(0),
        Universal( 0 ) ;

once
        begin
        if BandEdge <= 0 then
                RaiseRunTimeError( "BandEdge must be > zero" ) ;
        end ;

WhiteNoise = ( Close - Close[2] ) / 2 ;

// SuperSmoother Filter
a1 = ExpValue( -1.414 * 3.14159 / BandEdge ) ;
b1 = 2 * a1 * Cosine( 1.414 * 180 / BandEdge ) ;
c2 = b1 ;
c3 = -a1 * a1 ;
c1 = 1 - c2 - c3 ;
Filt = c1 * ( WhiteNoise + WhiteNoise [1] ) / 2 +
c2 * Filt[1] + c3 * Filt[2] ;
If Currentbar = 1 then
        Filt = 0 ;
If Currentbar = 2 then
        Filt = c1 * 0 * ( Close + Close[1] ) / 2 + c2 * Filt[1] ;
If Currentbar = 3 then
        Filt = c1 * 0 * ( Close + Close[1] ) / 2 + c2 * Filt[1] +
c3 * Filt[2] ;

// Automatic Gain Control (AGC)
Peak = .991 * Peak[1] ;
If Currentbar = 1 then
        Peak = .0000001 ;
If AbsValue( Filt ) > Peak then
        Peak = AbsValue( Filt ) ;
```

```
If Peak <> 0 then
        Universal = Filt / Peak ;

if Universal crosses over 0 then
        Buy next bar at Market ;

if Universal crosses under 0 then
        SellShort next bar at Market ;
```

A sample chart is shown in Figure 1.

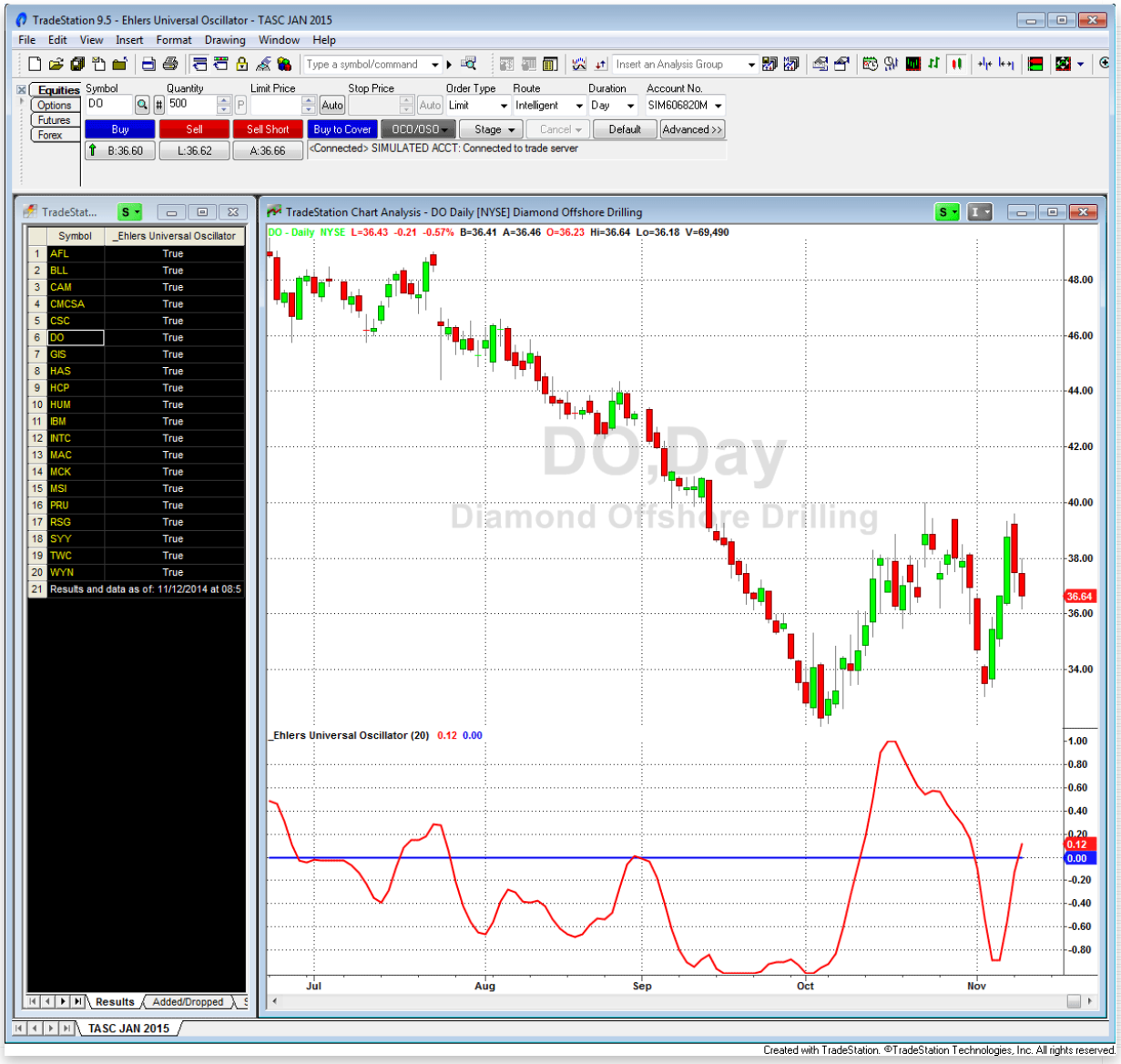


FIGURE 1: TRADESTATION, SCANNER. Here, the TradeStation Scanner shows a list of S&P 500 symbols that had the scan criteria met on 11/11/2014. The scan criteria is met when the universal oscillator crosses above or below zero. In the chart, the universal oscillator is applied to the symbol DO and displays the cross. The chart and the scanner are

linked such that clicking a scanner symbol name will force the chart to redraw with that symbol.

This article is for informational purposes. No type of trading or investment recommendation, advice, or strategy is being made, given, or in any manner provided by TradeStation Securities or its affiliates.

—Doug McCrary
TradeStation Securities, Inc.
www.TradeStation.com

BACK TO LIST

---

eSIGNAL: JANUARY 2015

For this month's Traders' Tip, we're providing the formula Universal_Oscillator.efs based on the formula given in John Ehlers' article in this issue, "Whiter Is Brighter."

The study contains formula parameters which may be configured through the *edit chart* window (right-click on the chart and select "edit chart"). A sample chart is shown in Figure 2.

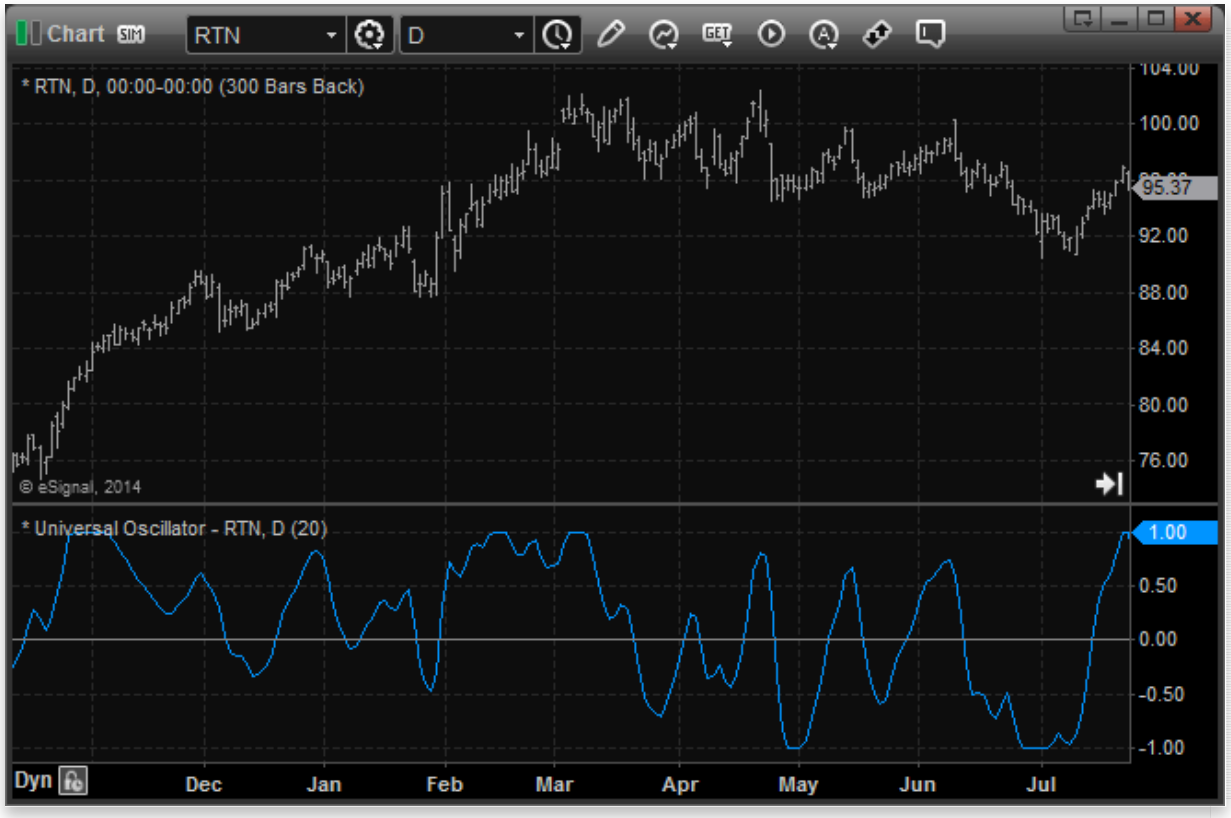


FIGURE 2: eSIGNAL. Here is an example implementation of the universal oscillator on Raytheon (RTN).

To discuss this study or download a complete copy of the formula code, please visit the EFS Library Discussion Board forum under the *forums* link from the support menu at www.esignal.com or visit our EFS KnowledgeBase at http://www.esignal.com/support/kb/efs/. The eSignal formula script (EFS) is also available for copying & pasting below.

```
/**********************************
Provided By:
    Interactive Data Corporation (Copyright © 2014)
    All rights reserved. This sample eSignal Formula Script (EFS)
    is for educational purposes only. Interactive Data Corporation
    reserves the right to modify and overwrite this EFS file with
    each new release.

Description:
    Universal Oscillator by John Ehlers



Formula Parameters:                     Default:
    BandEdge                            20

Version:            1.00  11/11/2014

Notes:
    The related article is copyrighted material. If you are not a subscriber
    of Stocks & Commodities, please visit www.traders.com.

**********************************/



var fpArray = new Array();



function preMain(){


    setStudyTitle("Universal Oscillator");
    setPriceStudy(false);



    setCursorLabelName("UniOsc", 0);



    addBand(0,PS_SOLID,1,Color.grey);

    var x = 0;

    fpArray[x] = new FunctionParameter("BandEdge", FunctionParameter.NUMBER);
```

```
    with(fpArray[x++]){
        setName("BandEdge");
        setLowerLimit(1);
        setDefault(20);
    }
}


var bVersion = null;

var WhiteNoise = 0;

var WhiteNoise_1 = 0;

var Filt = 0;

var Filt_1 = 0;

var Filt_2 = 0;

var Peak = 0;

var Peak_1 = 0;



function main(BandEdge){



    if (bVersion == null) bVersion = verify();
    if (bVersion == false) return;



    if(BandEdge==null) BandEdge = 20;



    var a1 = 0;

    var b1 = 0;

    var c1 = 0;

    var c2 = 0;

    var c3 = 0;

    var Universal = 0;



    if(getBarState()==BARSTATE_ALLBARS){

        xClose = close();

    }
```

```
    if(getBarState()==BARSTATE_NEWBAR){

        WhiteNoise_1 = WhiteNoise;

        Filt_2 = Filt_1;

        Filt_1 = Filt;

        Peak_1 = Peak;

    }



    WhiteNoise = (xClose.getValue(0)-xClose.getValue(-1))/2;

    a1 = Math.exp(-1.414*3.14159/BandEdge);

    b1 = 2*a1*Math.cos(1.414*180/BandEdge);

    c2 = b1;

    c3 = -a1*a1;

    c1 = 1-c2-c3;

    Filt = c1*(WhiteNoise+WhiteNoise_1)/2+c2*Filt_1+c3*Filt_2;



    if(getCurrentBarCount()==1) Filt = 0

    if(getCurrentBarCount()==2) Filt =
c1*0*(xClose.getValue(0)+xClose.getValue(-1))/2+c2*Filt_1;

    if(getCurrentBarCount()==3) Filt =
c1*0*(xClose.getValue(0)+xClose.getValue(-1))/2+c2*Filt_1+c3*Filt_2;



    Peak = .991*Peak_1;

    if(getCurrentBarCount()==1) Peak = 0.0000001;

    if(Math.abs(Filt)>Peak) Peak = Math.abs(Filt);

    if(Peak!=0) Universal = Filt/Peak;



    return Universal;

}



function verify(){
```

```
    var b = false;


    if (getBuildNumber() < 779){


        drawTextAbsolute(5, 35, "This study requires version 8.0 or later.",
            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,
            null, 13, "error");
        drawTextAbsolute(5, 20, "Click HERE to upgrade.@URL=http://www.esignal.com
/download/default.asp",
            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,
            null, 13, "upgrade");
        return b;
    }
    else
        b = true;

    return b;
}
```

—Eric Lippert
eSignal, an Interactive Data company
800 779-6555, www.eSignal.com

BACK TO LIST

METASTOCK: JANUARY 2015

John Ehlers' article in this issue, "Whiter Is Brighter," introduces his *universal oscillator*. The MetaStock code for this indicator follows:

```
Universal Oscillator:

bandedge:= 20;
whitenoise:= (C - Ref(C,-2))/2;

{super smoother filter}
a1:= Exp(-1.414 * 3.14159 / bandedge);
b1:= 2*a1 * Cos(1.414*180 /bandedge);
c2:= b1;
c3:= -a1 * a1;
c1:= 1 - c2 - c3;
filt:= c1 * (whitenoise + Ref(whitenoise, -1))/2 + c2*PREV + c3*Ref(PREV,-1);
filt1:= If(Cum(1) = 0, 0, If(Cum(1) = 2, c2*PREV,
If(Cum(1) = 3, c2*PREV + c3*Ref(PREV,-1), filt)));

pk:= If(Cum(1) = 2, .0000001,
If(Abs(filt1) > PREV, Abs(filt1), 0.991 * PREV));
denom:= If(pk=0, -1, pk);
```

```
If(denom = -1, PREV, filt1/pk)
```

—William Golson
MetaStock Technical Support
www.metastock.com

BACK TO LIST

WEALTH-LAB: JANUARY 2015

The *universal oscillator* introduced in John Ehlers' article in this issue, "Whiter Is Brighter," is an evolution of Ehlers' SuperSmoother filter, which was introduced in his January 2014 STOCKS & COMMODITIES article "Predictive And Successful Indicators"). The new indicator follows the short-term variations in price without introducing extra delay. It is comfortably controlled through just a single input — the *bandedge* — which basically is frequency. The smaller it is set, the less lag there is, but the smoother the indicator's outline will be. Built-in automatic gain control normalizes the output to vary between -1 to +1.

As an illustration of its application, a straightforward short-term countertrend system could, for example, use the following rules:

- Cover your short position and go long when the universal oscillator crosses below zero
- Sell and go short when the universal oscillator crosses above zero.

Since it is really a universal oscillator, the indicator can power up a trend-trading system just as easily:

- Buy when the long-term universal oscillator crosses above zero
- Sell when the long-term universal oscillator crosses below zero.

The trading system code we're providing for Wealth-Lab implements both approaches. To switch between trend- and countertrend trading modes, drag the parameter slider named "swing/trend" at the bottom left part of the screen, and Wealth-Lab will instantly rebuild the backtest.

Wouldn't it be interesting to adapt to different market conditions, distinguishing a trending market from a range-bound one? Motivated traders can take it a step further and automate the switching from trending to mean-reversion mode and vice versa. To achieve that, they can employ a dedicated trendiness oscillator like ADX, the Dreiss choppiness index, or McEwan's volatility switch [see Ron McEwan's article in the February 2013 STOCKS & COMMODITIES, "The Volatility (Regime) Switch Indicator," for a description].

To execute the example trading system, Wealth-Lab users need to install (or update) to the latest version of the TASCIndicators library from the *extensions* section of our website if they haven't already done so, and restart Wealth-Lab.

A sample chart is shown in Figure 3.

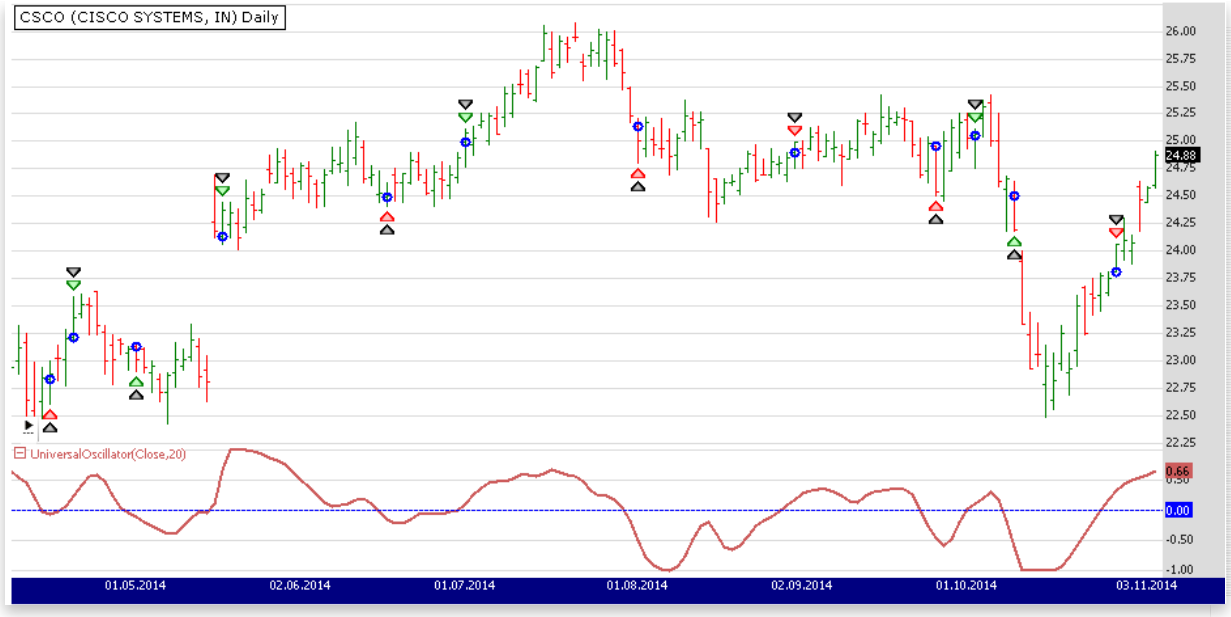


FIGURE 3: WEALTH-LAB. Here, a Wealth-Lab 6 chart illustrates application of the system's rules on a daily chart of Cisco Systems (CSCO).

C# Code

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Drawing;
using WealthLab;
using WealthLab.Indicators;
using TASCIndicators;

namespace WealthLab.Strategies
{
        public class UniversalOscSARStrategy : WealthScript
        {
                StrategyParameter be;
                StrategyParameter mode;

                public UniversalOscSARStrategy()
                {
                        be = CreateParameter("Band edge", 20, 5, 200, 5);
                        mode = CreateParameter("Swing/Trend", 0, 0, 1, 1);
                }

                protected override void Execute()
                {
                        bool swingTrade = mode.ValueInt == 0 ? true : false;
                        int period = be.ValueInt;

                        UniversalOscillator uo =
UniversalOscillator.Series(Close,period );
                        ChartPane uPane = CreatePane(30,false,true); HideVolume();
                        PlotSeries(uPane,uo,Color.IndianRed,LineStyle.Solid,2);
                        DrawHorzLine(uPane,0,Color.Blue,LineStyle.Dashed,1);
```

```
                        for(int bar = GetTradingLoopStartBar(period); bar <
Bars.Count; bar++)
                        {
                                // Detect crossover/crossunder and store state in
variable
                                bool uXo = CrossOver(bar, uo, 0);
                                bool uXu = CrossUnder(bar, uo, 0);

                                if( swingTrade ) // Countertrend trading
                                {
                                        // The first trade
                                        if (Positions.Count == 0){
                                                if ( uXu )
                                                        BuyAtMarket( bar + 1 );
                                                else if( uXo )
                                                        ShortAtMarket( bar + 1 );
                                        }
                                                // Subsequent trades
                                        else
                                        {
                                                Position p =
LastPosition;
                                                if ( p.PositionType ==
PositionType.Long )
                                                {
                                                        if ( uXo )
                                                        {
                                                                SellAtMarket( bar +
1, p );
                                                                ShortAtMarket( bar +
1 );
                                                        }
                                                }
                                                else if ( uXu )
                                                {
                                                        CoverAtMarket( bar + 1, p );
                                                        BuyAtMarket( bar + 1 );
                                                }
                                        }
                                }
                                else // Trend trading
                                {
                                        if (IsLastPositionActive)
                                        {
                                                if ( uXu )
                                                        SellAtMarket( bar + 1,
LastPosition );
                                        }
                                        else
                                        {
                                                if ( uXo )
                                                        BuyAtMarket( bar + 1 );
                                        }

}
                        }
                }
        }
}
```

—Eugene, Wealth-Lab team
MS123, LLC
www.wealth-lab.com

BACK TO LIST

AMIBROKER: JANUARY 2015

In “Whiter Is Brighter” in this issue, author John Ehlers presents his *universal oscillator*, which is based on filtering two-day price momentum using a second-order infinite impulse response (IIR) filter.

The code for the universal oscillator as implemented in AmiBroker formula language (AFL) is shown here:

```
LISTING 1.
// John F. Ehlers' Universal Oscillator
// January 2015 S&C Traders' Tips
//
// AFL implementation - TJ
SetBarsRequired( 100 );

BandEdge = Param("BandEdge", 20, 2, 100, 1 );

WhiteNoise = ( Close - Ref( Close, -2 ) ) / 2;

// SuperSmoother filter
// note: in AFL angles are in radians
pi = 3.1415926;
r2 = sqrt( 2 );
a1 = exp( - r2 * pi / BandEdge );
b1 = 2 * a1 * cos( r2 * pi / BandEdge );
c2 = b1;
c3 = - a1 * a1;
c1 = 1 - c2 - c3;

pk = 0.0000001;
universal = 0;

input = Nz( WhiteNoise + Ref( WhiteNoise, -1 ) ) / 2;
Filt = input;

// IIR+AGC loop
for( i = 2; i < BarCount; i++ )
{
 Filt[ i ] = c1 * input[ i ]  +
             c2 * Filt[ i - 1 ] +
             c3 * Filt[ i - 2 ];
```

```
 pk = 0.991 * pk;

 af = abs( Filt[ i ] );

 if( af > pk ) pk = af;

 universal[ i ] = Filt[ i ] / pk;
}

Plot( universal, "Universal" + _PARAM_VALUES(), colorRed, styleThick );
Plot( 0, "", colorBlue, styleNoLabel );
```

To use the oscillator, input the code into the formula editor and press *apply indicator*. A sample chart is shown in Figure 4.

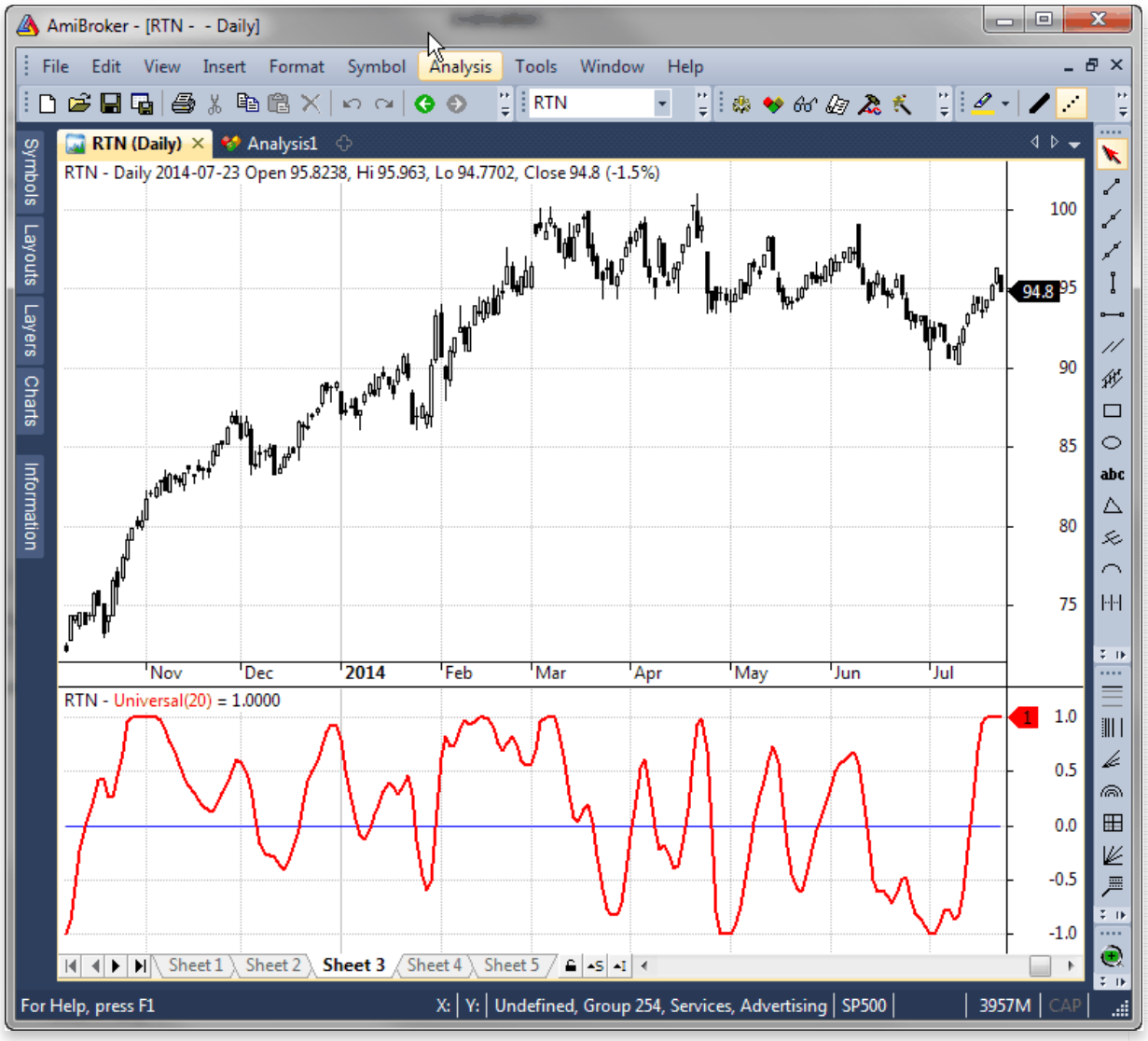


FIGURE 4: AMIBROKER. Here, a 20-day universal oscillator (bottom pane) is shown on a daily price chart of RTN, replicating the chart from John Ehlers' article in this issue.

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

BACK TO LIST

AIQ: JANUARY 2015

The AIQ code based on John Ehlers' article in this issue, "Whiter Is Brighter," is provided at www.TradersEdgeSystems.com/traderstips.htm. It is also shown below, for reference.

```
!WHITER IS BRIGHTER
!Author: John Ehlers, TASC Jan 2015
!Coded by: Richard Denning 11/11/2014
!www.TradersEdgeSystems.com

!Universal Oscillator
!(c) 2014 John F. Ehlers

!INPUTS:
BandEdge is 20.

!SuperSmoother Filter
WhiteNoise is ([Close] - val([Close],2)) / 2.
a1 is Exp(-1.414*3.14159 / BandEdge).
b1 is 2*a1*Cos((1.414*180) / BandEdge).
c2 is b1.
c3 is -a1*a1.
c1 is 1 - c2 - c3.

DaysInto is ReportDate() - RuleDate().
Stop if DaysInto > 10.
stopfilt is iff(stop,[close], filt).
Filt is c1*(WhiteNoise + valresult(WhiteNoise,1)) / 2
  + c2*valresult(stopfilt,1) + c3*valresult(stopfilt,2).
stopPeak is iff(stop,[close], peak).
Peak is 0.991*valresult(stopPeak,1).
 !Plot for Univeral Oscillator

Universal is (Filt / (max(Peak,abs(Filt))))*10.
```

—Richard Denning
info@TradersEdgeSystems.com
for AIQ Systems

BACK TO LIST

TRADERSSTUDIO: JANUARY 2015

The TradersStudio code based on John Ehlers' article in this issue, "Whiter Is Brighter," is provided at the following websites:

- www.TradersEdgeSystems.com/traderstips.htm
- www.TradersStudio.com → Traders Resources → Traders' Tips

The following code files are provided in the download:

- Function: EHLERS_UNIVERSAL_OSC — returns the oscillator values
- Indicator plot: EHLERS_UNIVERSAL_OSC_IND — plots the universal oscillator.

```
'WHITER IS BRIGHTER
'Author: John Ehlers, TASC Jan 2015
'Coded by: Richard Denning 11/11/2014
'www.TradersEdgeSystems.com

'Universal Oscillator
'(c) 2014 John F. Ehlers
Function EHLERS_UNIVERSAL_OSC(BandEdge)
    Dim WhiteNoise As BarArray
    Dim a1
    Dim b1
    Dim c1
    Dim c2
    Dim c3
    Dim Filt As BarArray
    Dim Peak As BarArray
    Dim Universal

    If BarNumber=FirstBar Then
        'BandEdge = 20
        WhiteNoise = 0
        a1 = 0
        b1 = 0
        c1 = 0
        c2 = 0
        c3 = 0
        Filt = 0
        Peak = 0
        Universal = 0
    End If

  'SuperSmoother Filter
    WhiteNoise = (Close - Close[2]) / 2
    a1 = Exp(-1.414*3.14159 / BandEdge)
    b1 = 2*a1*Cos(1.414*180 / BandEdge)
    c2 = b1
    c3 = -a1*a1
```

```
    c1 = 1 - c2 - c3
    Filt = c1*(WhiteNoise + WhiteNoise [1]) / 2 + c2*Filt[1] + c3*Filt[2]
    If CurrentBar - 1 = 1 Then
        Filt = 0
    End If
    If CurrentBar - 1 = 2 Then
        Filt = c1*0*(Close + Close[1])/2 + c2*Filt[1]
    End If
    If CurrentBar - 1 = 3 Then
        Filt = c1*0*(Close + Close[1])/2 + c2*Filt[1] + c3*Filt[2]
    End If
    Peak = .991*Peak[1]
    If CurrentBar - 1 = 1 Then
        Peak = .0000001
    End If
    If Abs(Filt) > Peak Then
        Peak = Abs(Filt)
    End If
    If Peak <> 0 Then
        Universal = Filt / Peak
    End If
EHLERS_UNIVERSAL_OSC = Universal
End Function
'-----------------------------------------------------------------------------
' Indicator Plot:
Sub EHLERS_UNIVERSAL_OSC_IND(BandEdge)
plot1(EHLERS_UNIVERSAL_OSC(BandEdge))
plot2(0)
End Sub
'-----------------------------------------------------------------------------
```

In Figure 5, I show a chart of the full-sized S&P 500 futures contract (using data from Pinnacle Data Corp.) with the universal oscillator.

FIGURE 5: TRADERSSTUDIO. The universal oscillator

is plotted on the S&P 500 futures.

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

NEUROSHELL TRADER: JANUARY 2015

The *universal oscillator* introduced by John Ehlers in his article in this issue, "Whiter Is Brighter," can be implemented in NeuroShell Trader using NeuroShell Trader's ability to call external dynamic linked libraries. Dynamic linked libraries can be written in C, C++, Power Basic, or Delphi.

After moving the EasyLanguage code provided in Ehlers' article to your preferred compiler and creating a DLL, you can insert the resulting indicators as follows:

1. Select *new indicator* from the Insert menu
2. Choose the "external program & library calls" category
3. Select the appropriate external DLL call indicator
4. Set up the parameters to match your DLL
5. Select the *finished* button.

Dynamic trading systems can be easily created in NeuroShell Trader by combining the universal oscillator with NeuroShell Trader's genetic optimizer to find optimal lengths. Similar filter and cycle-based strategies can also be created using indicators found in John Ehlers' Cybernetic and MESA91 NeuroShell Trader add-ons. A sample chart is shown in Figure 6.

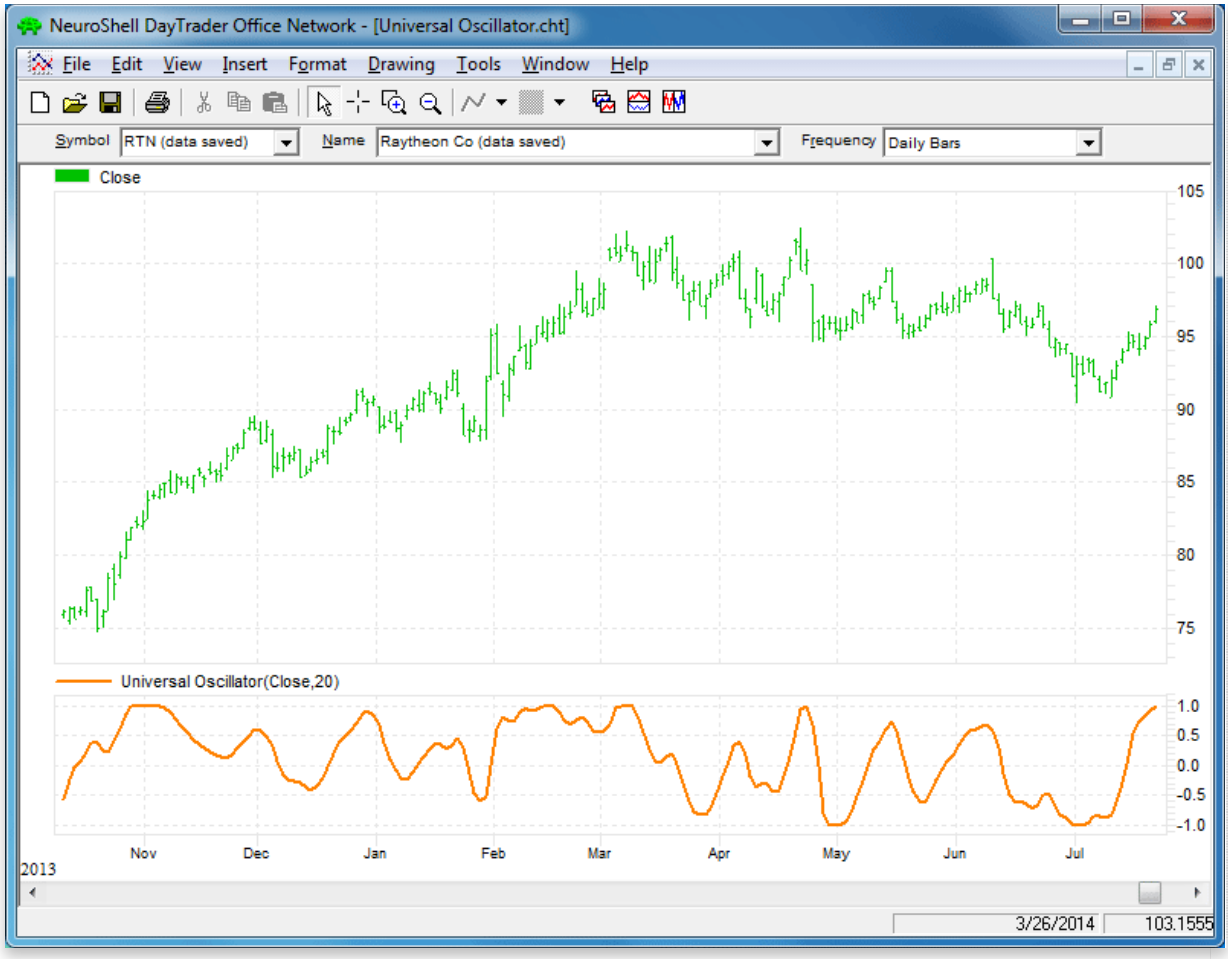


FIGURE 6: NEUROSHELL TRADER. This NeuroShell Trader chart displays the ultimate oscillator indicator

Users of NeuroShell Trader can go to the STOCKS & COMMODITIES section of the NeuroShell Trader free technical support website to download a copy of this or any previous Traders' Tips.

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

BACK TO LIST

NINJATRADER: JANUARY 2015

In "Whiter Is Brighter" in this issue, author John Ehlers introduces his universal oscillator. We're making the UniversalOsc indicator, based on his article, available for download at www.ninjatrader.com/SC/January2015SC.zip.

Once it has been downloaded, from within the NinjaTrader Control Center window, select the menu *File → Utilities → Import NinjaScript* and select the downloaded file. This file is for

NinjaTrader version 7 or greater.

You can review the indicator source code by selecting the menu *Tools → Edit NinjaScript → Indicator* from within the NinjaTrader Control Center window and selecting the *UniversalOsc* file.

A sample chart implementing the strategy is shown in Figure 7.

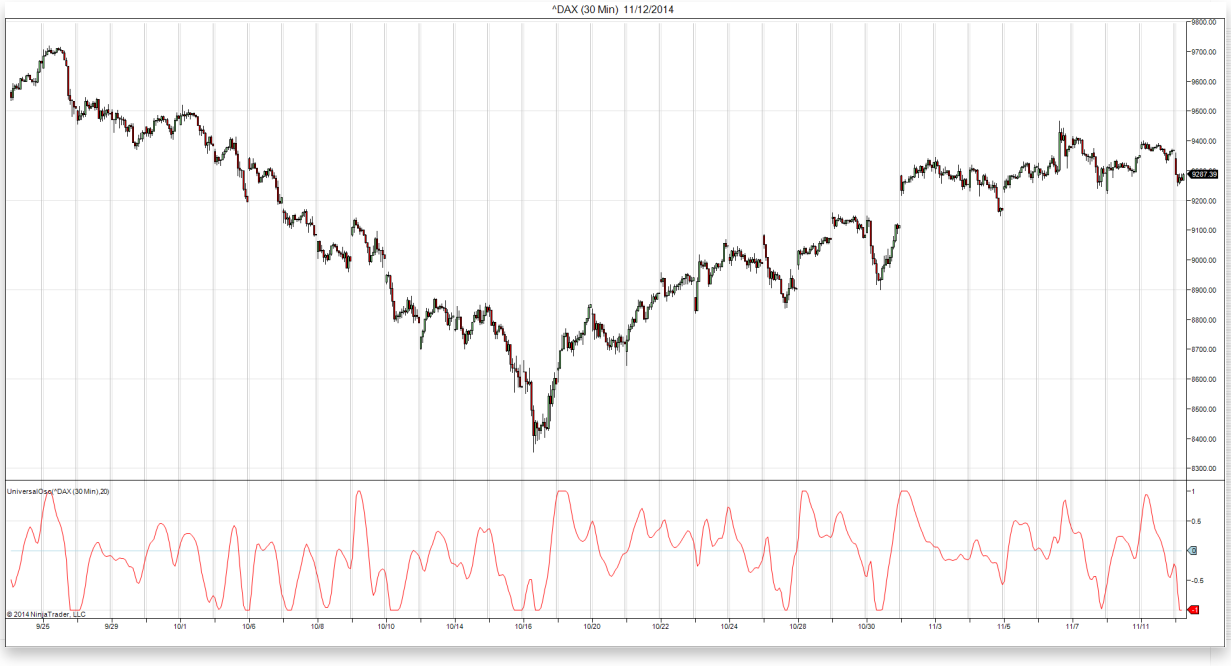


FIGURE 7: NINJATRADER. This screenshot shows the indicator applied to a 30-minute chart of the DAX index.

—Raymond Deux & Jesse Nelson, NinjaTrader, LLC
www.ninjatrader.com

BACK TO LIST

UPDATA: JANUARY 2015

Our Traders' Tip this month is based on John Ehlers' article in this issue, "Whiter Is Brighter." In it, Ehlers seeks to develop a zero-lag oscillator in order to generate signals on underlying market data that is said to display "pink noise" properties. This is distinct from white noise in that some memory is inherent in the data, and could form the basis of a short-term trading model.

The Updata code for this article is in the Updata Library and may be downloaded by clicking the

custom menu and *indicator* library. Those who cannot access the library due to a firewall may paste the code shown below into the Updata custom editor and save it.

```
'Universal Oscillator
DISPLAYSTYLE 2LINES
INDICATORTYPE CHART
INDICATORTYPE2 TOOL
COLOUR2 RGB(200,0,0)
PARAMETER "Band Edge" @BANDEDGE=20
NAME Universal Oscillator
@WHITENOISE=0
@A1=0
@B1=0
@C1=0
@C2=0
@C3=0
@FILT=0
@PEAK=0
@UNIVERSAL=0
@ROOT2=0
@PI=0
FOR #CURDATE=2 TO #LASTDATE
   @ROOT2=EXPBASE(2,0.5)
   @PI=CONST_PI
   @WHITENOISE=(CLOSE-CLOSE(2))/2
   @A1=EXP(-@ROOT2*@PI/@BANDEDGE)
   @B1=2*@A1*COS(@ROOT2*@PI*0.5/@BANDEDGE)
   @C2=@B1
   @C3=-@A1*@A1
   @C1=1-@C2-@C3
   @FILT=@C1*0.5*(@WHITENOISE+HIST(@WHITENOISE,1))+@C2*HIST(@FILT,1)+@C3*HIST(@FILT,2)
   IF #CURDATE=2
      @FILT=0
   ELSEIF #CURDATE=3
      @FILT=@C2*HIST(@FILT,1)
   ELSEIF #CURDATE=4
      @FILT=@C2*HIST(@FILT,1)+@C3*HIST(@FILT,2)
   ENDIF
   'AUTOMATIC GAIN CONTROL (AGC)
   @PEAK=0.991*HIST(@PEAK,1)
   IF #CURDATE=2
      @PEAK=0.0000001
   ENDIF
   IF ABS(@FILT)>@PEAK
      @PEAK=ABS(@FILT)
   ENDIF
   IF @PEAK!=0
      @UNIVERSAL=@FILT/@PEAK
   ENDIF
   @PLOT=@UNIVERSAL
   @PLOT2=0
NEXT
```

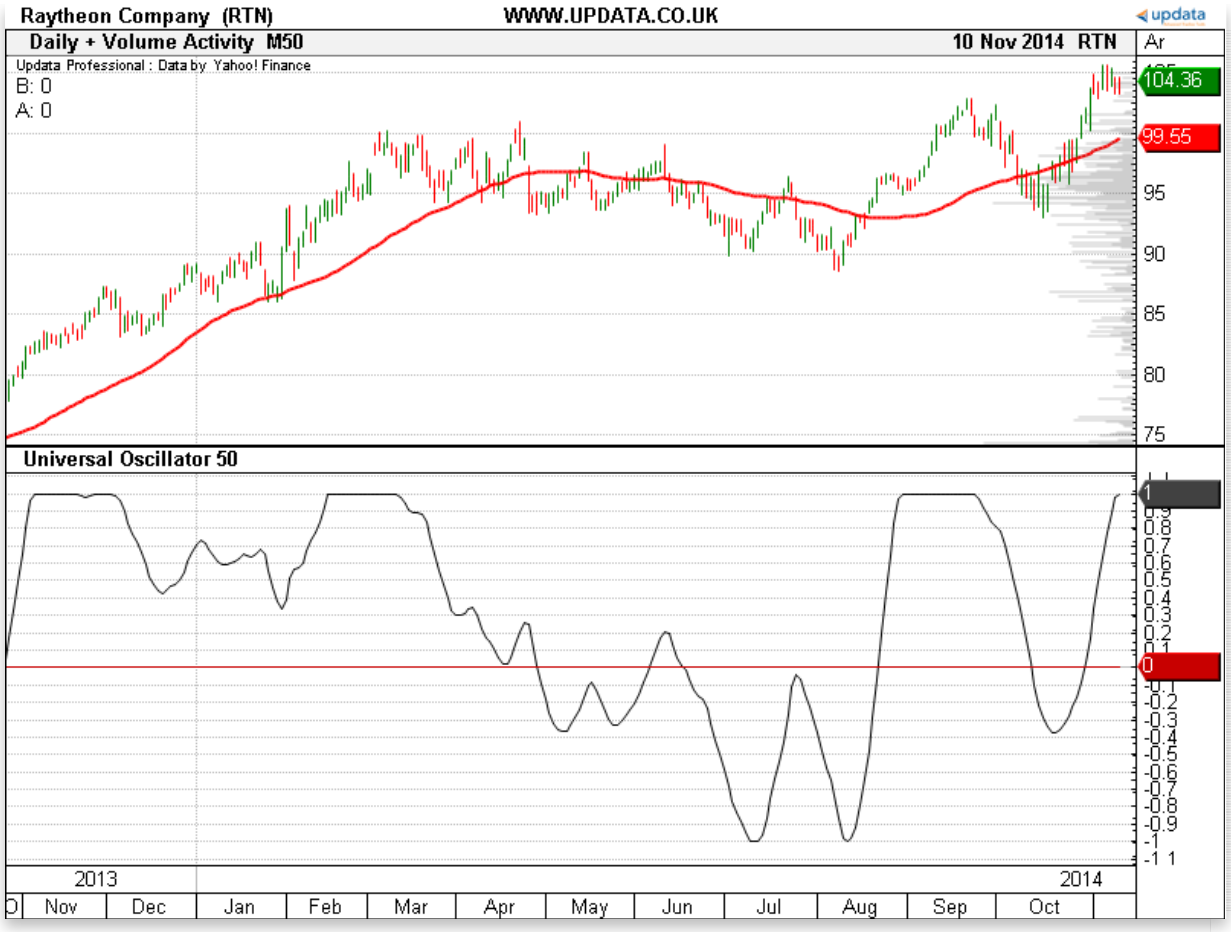


FIGURE 8: UPDATA. This chart shows the universal oscillator of band edge 50, as applied to Raytheon Company in daily resolution data.

—Updata support team
support@updata.co.uk, www.updata.co.uk

BACK TO LIST

VT TRADER: JANUARY 2015

This Traders' Tip is based on "Whiter Is Brighter" in this issue by John Ehlers. We'll be offering the universal oscillator for download in our VT client forums at http://forum.vtsystems.com along with other precoded and free indicators and trading systems.

The VT Trader instructions for creating the aforementioned indicator are as follows:

1. VT Trader's Ribbon→Technical Analysis menu→Indicators group→Indicators Builder→[New] button
2. In the General tab, type the following text into each corresponding text box:

```
Name: TASC - 01/2015 - Universal Oscillator
Function Name Alias: tasc_UniOsc
Label Mask: TASC - 01/2015 - Universal Oscillator (%BandEdge%) = %Universal%
Placement: New Frame
Data Inspection Alias: Universal Oscillator
```

3. In the Input Variable(s) tab, create the following variables:

```
[New] button...
Name: BandEdge
Display Name: Band Edge Periods
Type: integer
Default: 20
```

4. In the Output Variable(s) tab, create the following variables:

```
[New] button...
Var Name: Universal
Name: (Universal)
Line Color: dark blue
Line Width: 2
Line Type: solid line
```

5. In the Horizontal Line tab, create the following horizontal lines:

```
[New] button...
Value: 0
Line Color: black
Line Width: thin
Line Type: dashed
```

6. In the Formula tab, copy and paste the following formula:

```
{Provided By: Visual Trading Systems, LLC}
{Copyright: 2014}
{Description: TASC, January 2015 - "Relax, It's Just Noise - Whiter Is Brighter"
by John Ehlers}
{File: tasc_UniOsc.vtscr - Version 1.0}

Whitenoise:= If(BarCount()<3,0,(Close - Ref(Close,-2)) / 2);

// SuperSmoother Filter

a1:= If(BarCount()=1, 0, Exp(-1.414*3.14159 / Bandedge));
b1:= If(BarCount()=1, 0, 2*a1*Cos(1.414*180 / Bandedge));
c2:= If(BarCount()=1, 0, b1);
c3:= If(BarCount()=1, 0, -a1*a1);
c1:= If(BarCount()=1, 0, 1 - c2 - c3);
Filt:= If(BarCount()=1, 0,
       If(BarCount()=2, c1*0*(Close + Ref(Close,-1))/2 + c2*Prev(0),
       If(BarCount()=3, c1*0*(Close + Ref(Close,-1))/2 + c2*Prev(0) +
c3*Ref(Prev(0),-1),
       c1*(Whitenoise + Ref(Whitenoise,-1)) / 2 + c2*Prev(0) + c3*Ref(Prev(0),-
1))));
```

```
// automatic Gain Control (aGC)

_Peak:= If(BarCount()=1, 0.0000001,
        If(Abs(Filt) > Prev(0), Abs(Filt),
        0.991*Prev(0)));

Universal:= Filt / _Peak;
```

7. Click the “Save” icon in the toolbar to finish building the Universal Oscillator.

A sample chart implementation is shown in Figure 9.

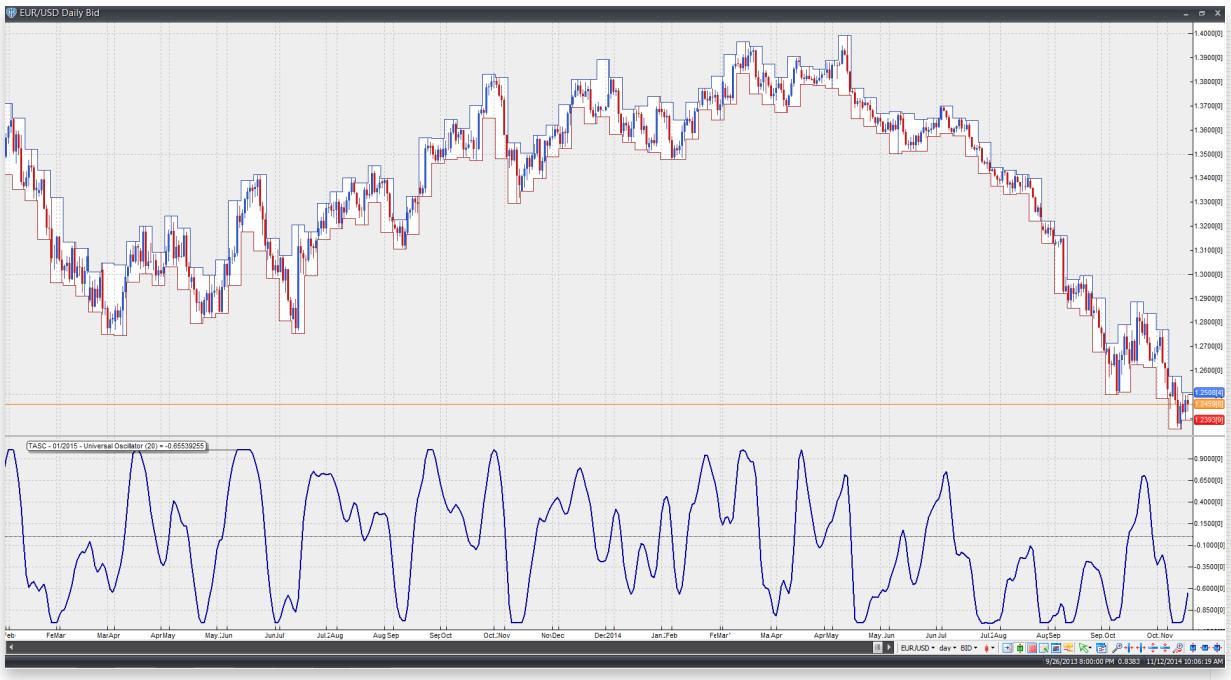


FIGURE 9: VT TRADER. Here, the universal oscillator is shown on a daily candle chart of EUR/USD.

Risk disclaimer: Past performance is not indicative of future results. Forex trading involves a substantial risk of loss and may not be suitable for all investors.

—Visual Trading Systems, LLC
vttrader@vtsystems.com, www.vtsystems.com

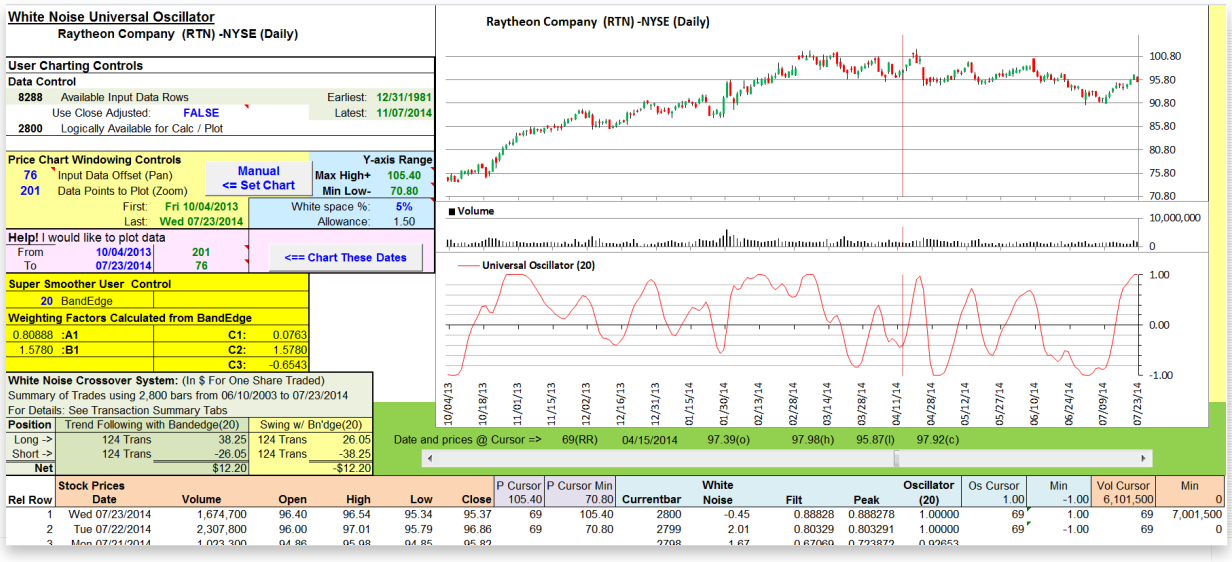
BACK TO LIST

MICROSOFT EXCEL: JANUARY 2015

John Ehlers’ *white noise universal oscillator*, introduced in his article in this issue (“Whiter Is Brighter”), is elegant in its computational simplicity. Four columns of Excel cell formulas do the job nicely.

And, even better, this oscillator is remarkable in my experience for its minimal (zero?) signal-to-response lag.

The chart in Figure 10 is my approximation of Figure 3 from Ehlers' article. Setting *bandedge* to 50 using the dates as shown here will replicate Figure 4 from Ehlers' article.



White Noise Universal Oscillator
Raytheon Company (RTN) -NYSE (Daily)

User Charting Controls
Data Control
8288 Available Input Data Rows — Earliest: 12/31/1981
Use Close Adjusted: FALSE — Latest: 11/07/2014
2800 Logically Available for Calc / Plot

Price Chart Windowing Controls
76 Input Data Offset (Pan)
201 Data Points to Plot (Zoom)
First: Fri 10/04/2013
Last: Wed 07/23/2014

Y-axis Range
Max High+ 105.40
Min Low- 70.80
White space %: 5%
Allowance: 1.50

Help! I would like to plot data
From 10/04/2013 201
To 07/23/2014 76

Super Smoother User Control
20 BandEdge

Weighting Factors Calculated from BandEdge
0.80888 :A1 — C1: 0.0763
1.5780 :B1 — C2: 1.5780
C3: -0.6543

White Noise Crossover System: (In $ For One Share Traded)
Summary of Trades using 2,800 bars from 06/10/2003 to 07/23/2014
For Details: See Transaction Summary Tabs

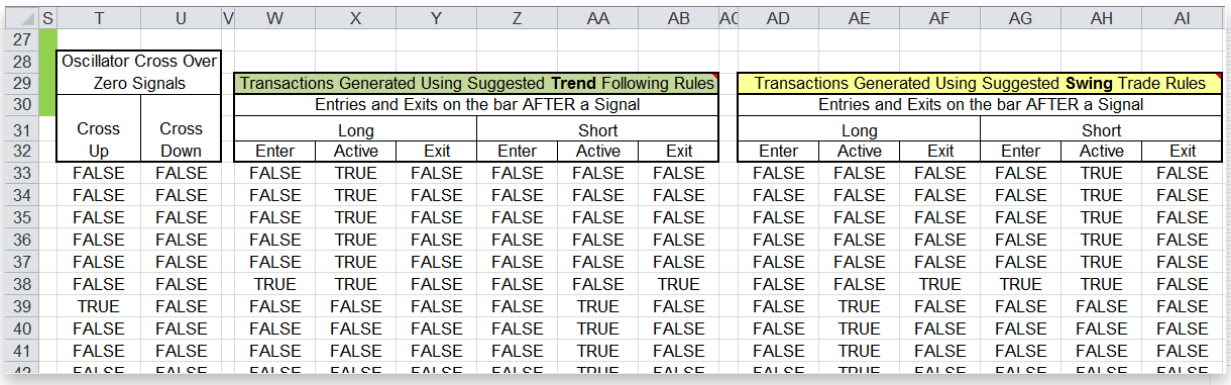
Position	Trend Following with Bandedge(20)		Swing w/ Bn'dge(20)	
Long ->	124 Trans	38.25	124 Trans	26.05
Short ->	124 Trans	-26.05	124 Trans	-38.25
Net		$12.20		-$12.20

Date and prices @ Cursor => 69(RR) 04/15/2014 97.39(o) 97.98(h) 95.87(l) 97.92(c)

Rel Row	Stock Prices Date	Volume	Open	High	Low	Close	P Cursor 105.40	P Cursor Min 70.80	Currentbar	White Noise	Filt	Peak	Oscillator (20)	Os Cursor 1.00	Min -1.00	Vol Cursor 6,101,500	Min 0
1	Wed 07/23/2014	1,674,700	96.40	96.54	95.34	95.37	69	105.40	2800	-0.45	0.88828	0.888278	1.00000	69	1.00	69	7,001,500
2	Tue 07/22/2014	2,307,800	96.00	97.01	95.79	96.86	69	70.80	2799	2.01	0.80329	0.803291	1.00000	69	-1.00	69	0
3	Mon 07/21/2014	1,023,300	94.86	95.98	94.85	95.82			2798	1.67	0.67069	0.723872	0.92653				

FIGURE 10: EXCEL, BANDEDGE. Here's an example price chart with bandedge set to the suggested swing trade value of 20.

You could also apply the concept to a countertrend trading strategy using the inverse of the rules proposed for a trend-following strategy. The results summarized in the lower left of Figure 10 reflect this inverse relationship and allow you to see how the two trading strategies fare side by side at any given bandedge setting.



	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI
27																	
28		Oscillator Cross Over															
29		Zero Signals			Transactions Generated Using Suggested **Trend** Following Rules							Transactions Generated Using Suggested **Swing** Trade Rules					
30					Entries and Exits on the bar AFTER a Signal							Entries and Exits on the bar AFTER a Signal					
31		Cross	Cross			Long			Short				Long			Short	
32		Up	Down		Enter	Active	Exit	Enter	Active	Exit		Enter	Active	Exit	Enter	Active	Exit
33		FALSE	FALSE		FALSE	TRUE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
34		FALSE	FALSE		FALSE	TRUE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
35		FALSE	FALSE		FALSE	TRUE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
36		FALSE	FALSE		FALSE	TRUE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
37		FALSE	FALSE		FALSE	TRUE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
38		FALSE	FALSE		TRUE	TRUE	FALSE	FALSE	FALSE	TRUE		FALSE	FALSE	TRUE	TRUE	TRUE	FALSE
39		TRUE	FALSE		FALSE	FALSE	FALSE	FALSE	TRUE	FALSE		FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
40		FALSE	FALSE		FALSE	FALSE	FALSE	FALSE	TRUE	FALSE		FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
41		FALSE	FALSE		FALSE	FALSE	FALSE	FALSE	TRUE	FALSE		FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
42		FALSE	FALSE		FALSE	FALSE	FALSE	FALSE	TRUE	FALSE		FALSE	TRUE	FALSE	FALSE	FALSE	FALSE

FIGURE 11: EXCEL, LOGIC TABLES. Trading logic tables are located just to the right of the charts.

Figure 11 shows the decision tables reflecting the two strategies using the current bandedge setting. Because this system is computed on bar close, these tables simulate trade execution on

the bar following a buy or sell crossover signal. The tables drive the trading details shown on the corresponding transaction summary tab in Figure 12. There is one summary tab for each strategy so that we may see in detail how long and short positions fare for that strategy.

Trend Following Transaction Summary (using trade size of 1 share)

2800 Logically Available Bars on the "CalculationsAndCharts" tab constitute a Back Test Transaction reporting period extending from 06/10/2003 to 07/23/2014

Summary for **LONG** trades					
124	LONG trades opened in this time period				
56	Winners	Biggest Win	10.65	Sum of Gains	117.76
68	Losers	Biggest Loss	-5.60	Sum of Losses	-79.51
				NET change for Period	38.25
				Smallest Long Account Balance	-3.50
			28	Open Trade; Marked to Current Bar Open $	

Enter Long			Exit Long			Trade	Running
Rel Row	Date	$	Rel Row	Date	$	Result	Balance
6	07/16/2014	95.02	OPEN	07/23/2014	96.40	1.38	38.25
38	05/30/2014	96.68	28	06/13/2014	95.70	-0.98	36.87
52	05/09/2014	97.35	45	05/20/2014	95.29	-2.06	37.85
66	04/21/2014	101.50	60	04/29/2014	95.90	-5.60	39.91
77	04/03/2014	100.30	73	04/09/2014	97.59	-2.71	45.51
118	02/04/2014	91.75	85	03/24/2014	97.88	6.13	48.22
132	01/14/2014	90.40	123	01/28/2014	88.47	1.93	42.09

Summary for **SHORT** trades					
124	Short trades opened in this time period				
34	Winners	Biggest Win	14.12	Sum of Gains	70.06
90	Losers	Biggest Loss	-3.91	Sum of Losses	-96.11
				NET change for Period	-26.05
				Smallest Short Account Balance	-26.73
			6		

Enter Short			Exit Short			Trade	Running
Rel Row	Date	$	Rel Row	Date	$	Result	Balance
28	06/13/2014	95.70	6	07/16/2014	95.02	0.68	-26.05
45	05/20/2014	95.29	38	05/30/2014	96.68	-1.39	-26.73
60	04/29/2014	95.90	52	05/09/2014	97.35	-1.45	-25.34
73	04/09/2014	97.59	66	04/21/2014	101.50	-3.91	-23.89
85	03/24/2014	97.88	77	04/03/2014	100.30	-2.42	-19.98
123	01/28/2014	88.47	118	02/04/2014	91.75	-3.28	-17.56
136	01/08/2014	88.68	132	01/14/2014	90.40	1.72	14.28

Notes | InputPriceData | Follow | CalculationsAndCharts | **Trend Transaction Summary** | Swing Transaction Summary

FIGURE 12: EXCEL, TRANSACTION TABS. Here's an example of the detailed transaction summary tabs.

The spreadsheet file for this Traders' Tip can be downloaded here. To successfully download it, follow these steps:

- Right-click on the Excel file link, then
- Select "save as" (or "save target as") to place a copy of the spreadsheet file on your hard drive.

—Ron McAllister, Excel and VBA programmer
rpmac_xltt@sprynet.com

BACK TO LIST

THINKORSWIM: JANUARY 2015

In "Whiter Is Brighter" by John Ehlers in this issue, Ehlers discusses eliminating the "noise" in the market to get a clearer picture of entry & exit points. From this we derived the UniversalOscillator strategy using our proprietary scripting language, thinkscript. To load the strategy into thinkorswim, simply click on the following link: http://tos.mx/zzhBRD and choose "Backtest in thinkorswim." Then choose to rename your study to "UniversalOscillatorStrategy." You can adjust the parameters within the *edit studies* window to fine tune your variables.

You can see from the chart in Figure 13 that the strategy on thinkorswim charts will give you entry points (the two blue arrows) when the flag is beginning and when the point forms. You can also see that the peak of the pole is the indicator with the exit point.

FIGURE 13: THINKORSWIM

To see how this strategy performed, simply right-click on the exit point and choose *show report*.

For a detailed description of the strategy or to read the full article, STOCKS & COMMODITIES subscribers can read Ehlers' full article in the article archive at Traders.com by clicking here.

—thinkorswim
A division of TD Ameritrade, Inc.
www.thinkorswim.com

BACK TO LIST

Originally published in the Jan 2015 issue of
Technical Analysis of STOCKS & COMMODITIES magazine.

February 2015

For this month's Traders' Tips, the focus is Dave Cline's article in this issue, "Candlesticks, Condensed." Here, we present the February 2015 Traders' Tips code with possible implementations in various software.

Code in the Python language is already provided by Cline, which S&C subscribers will find in the Subscriber Area of our website here.

The Traders' Tips section is provided to help the reader implement a selected technique from an article in this issue or another recent issue. The entries here are contributed by software developers or programmers for software that is capable of customization.

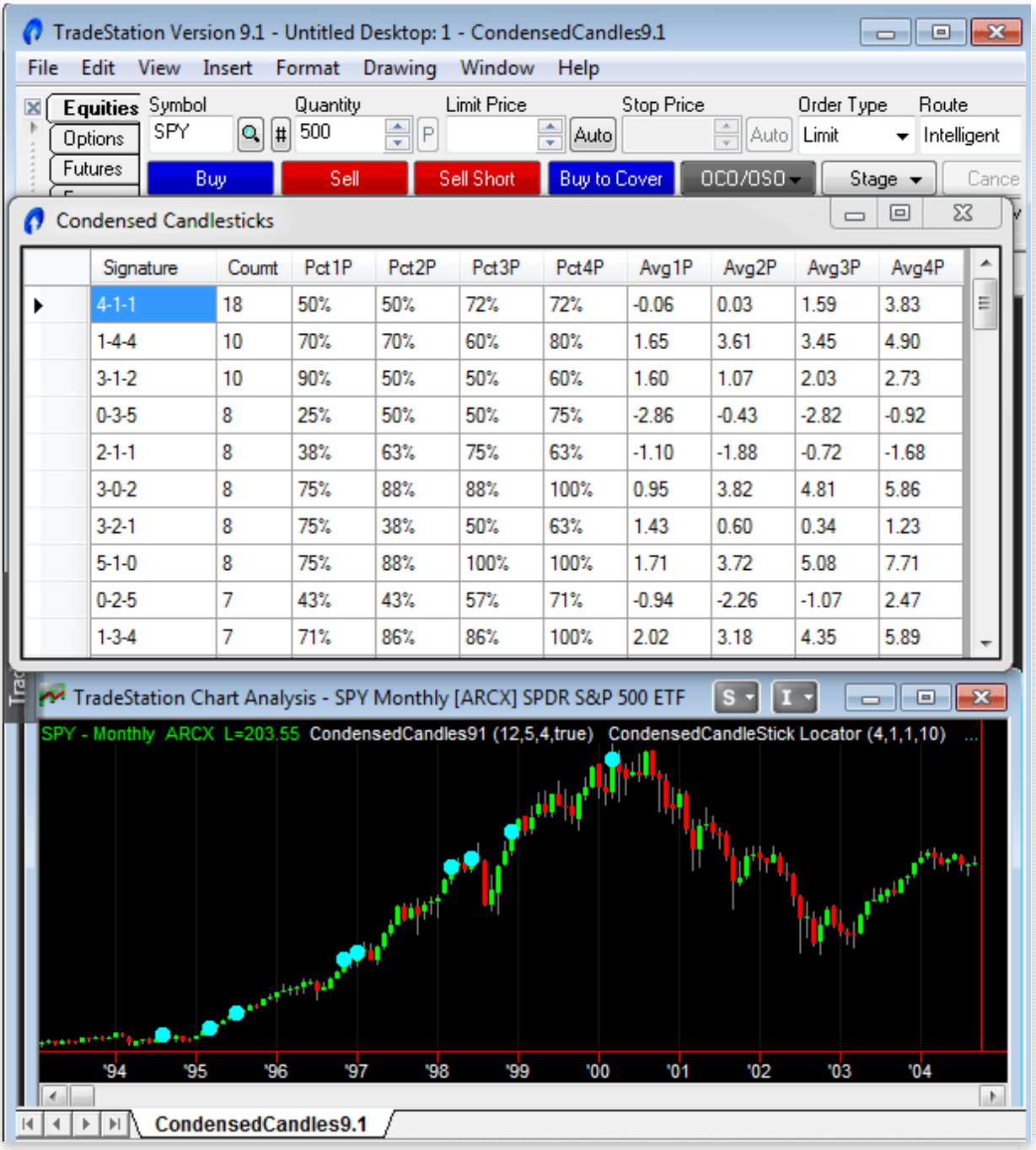
TRADESTATION: FEBRUARY 2015

In "Candlesticks, Condensed" in this issue, author Dave Cline describes a process for condensing every candle on a chart into three numbers: HO (high-open), HC (high-close) and OL (open-low). Each number is a fraction of the average daily range. The three numbers together (HO:HC:OL) make up the bar's signature. We are providing two indicators based on this process. The first calculates the signature and reproduces Cline's statistical summary. The code demonstrates the use of vectors and forms. The second locates bars with a desired input signature on the chart under study.

To download the EasyLanguage code, please visit our TradeStation and EasyLanguage support forum. The code from this article can be found here: http://www.tradestation.com/TASC-2015. The ELD filename is "_TASC_CondensedCandlesticks.ELD."

For more information about EasyLanguage in general, please see: http://www.tradestation.com/EL-FAQ.

A sample chart demonstrating these two indicators is shown in Figure 1.



Signature	Count	Pct1P	Pct2P	Pct3P	Pct4P	Avg1P	Avg2P	Avg3P	Avg4P
4-1-1	18	50%	50%	72%	72%	-0.06	0.03	1.59	3.83
1-4-4	10	70%	70%	60%	80%	1.65	3.61	3.45	4.90
3-1-2	10	90%	50%	50%	60%	1.60	1.07	2.03	2.73
0-3-5	8	25%	50%	50%	75%	-2.86	-0.43	-2.82	-0.92
2-1-1	8	38%	63%	75%	63%	-1.10	-1.88	-0.72	-1.68
3-0-2	8	75%	88%	88%	100%	0.95	3.82	4.81	5.86
3-2-1	8	75%	38%	50%	63%	1.43	0.60	0.34	1.23
5-1-0	8	75%	88%	100%	100%	1.71	3.72	5.08	7.71
0-2-5	7	43%	43%	57%	71%	-0.94	-2.26	-1.07	2.47
1-3-4	7	71%	86%	86%	100%	2.02	3.18	4.35	5.89

FIGURE 1: TRADESTATION. This depicts the indicator CondensedCandlesticks running on a monthly chart of SPY that starts with the February 1993 bar. The indicator generates the above info-grid displaying the most common CondensedCandlesticks. The signature column displays HO:HC:OL values. To display the bars with a specific signature, the indicator CondensedCandlestickLocator is provided. In this case, we are displaying the signature 4:1:1. The desired signature bars are noted with a cyan dot.n

_TASC_CONDENSEDCANDLES.ELD

```
_CondensedCandles91 (Indicator)
{
CondensedCandles is based upon the Article "Candlesticks, Condensed" by David Cline
appearing in the February 2015 issue of Technical Analysis of Stocks & Commodities

This indicator is designed to output the grid shown in Figure 6 of the article.

Intrabar Ticks should be TURNED OFF on the Indicator
}

using elsystem ;
using elsystem.collections ;
using elsystem.windows ;
using elsystem.windows.forms ;

inputs:
        iAvgRngLen( 12 ),
        iSegmentCount( 5 ),
        iMos2MosCount( 4 ) ;

variables:
        { BarsData is a dictionary with the key being the BarNumber.ToString and the
          value being BarData vector corresponding to that bar }
        Dictionary BarsData( NULL ),

        {
        The BarData vector simulates a class to hold specific data about a bar
        in specific elements using the bv... constants as indices.  There is a
        BarData vector in the BarsData dictionary for each bar we process
        }
        Vector BarData( NULL ),
        Vector PatternBarData( NULL ),
        Vector SortVector( NULL ),
        {
        Patterns is a dictionary with the key being the signature and the value
        being a dictionary with specific named items to hold all the needed
        calculated values for the signature.

                The Names slots include:
                Signature               - Repeats signature
                Count                   - Count of number of bars with the signature
                PctCount<n>             - Count of number of occurences of <n>-month
                PctUpCount<n>   - Count of number of Up occurences of <n>-month
                AvgCumm<n>              - n-month cummulative gain/loss
        }
        Dictionary Patterns( NULL ),

        { Dictionary of actual pattern data - used for current pattern }
        Dictionary PatternData( NULL ),

        SegmentDivisor( 0.0 ),
        AverageRange( 0.0 ),
```

```
        CandleRange( 0.0 ),
        RangeMultiplier( 0.0 ),
        HO( 0.0 ),
        HC( 0.0 ),
        OL( 0.0 ),
        CandleSignature( "" ),
        int M2MBack( 0 ),
        M2MBackStr( "" ),
        intrabarpersist int PatternsCount( 0 ),
        PatternBarNo( 0 ),
        PatternBarNoStr( "" ),
        int PatternIndex( 0 ) ;

variables:
        form FGrid( NULL ),
        Panel Pnl( NULL ),
        DataGridView Grd( NULL ),
        DataGridViewColumn Col( NULL ),
        DataGridViewRow Row( NULL ) ;

constants:
        bsOpening( 0 ),
        bsWithin( 1 ),
        bsClosing( 2 ),

        { BarData Element Indices }
        bvNumber( 0 ),
        bvOpen( 1 ),
        bvHigh( 2 ),
        bvLow( 3 ),
        bvClose( 4 ),
        bvAvgRange( 5 ),
        bvSignature( 6 ),
        bvLast( 6 ) ;


{ Initialize - Create Dictionaries, check input parameters and initialize any
  other variables }
method void AnalysisTechnique_Initialized( Object sender,
 InitializedEventArgs args )
        begin
        try
                Patterns = Dictionary.Create() ;
                BarsData = Dictionary.Create() ;
                SortVector = Vector.Create() ;

                if iAvgRngLen < 6 then
                        begin
                        Alert( "iAvgRngLen less than 6" ) ;
                        Abort ;
                        end ;

                if iSegmentCount < 3 then
                        begin
                        Alert( "iSegmentCount less than 3" ) ;
                        Abort ;
                        end ;

                if iMos2MosCount < 1 or iMos2MosCount > 12 then
                        begin
```

```
                        Alert( "iMos2MosCount is not between 1 and 12, inclusive" ) ;
                        Abort ;
                        end ;

                SegmentDivisor = 100.0 / iSegmentCount ;

                { Build form and Grid in form to show results }
                FGrid = form.Create( "Condensed Candlesticks", 800, 800 ) ;

                Grd = DataGridView.Create() ;
                Grd.AllowUserToAddRows = false ;
                Grd.Location( 0, 0 ) ;
                Grd.Dock = DockStyle.Fill ;
                FGrid.AddControl( Grd ) ;

                { add columns }
                Grd.Columns.Add( "Signature" ) ;
                Grd.columns[0].Width = 75 ;

                Grd.Columns.Add( "Coumt" ) ;
                Grd.columns[1].Width = 45 ;

                for M2MBack = 1 TO iMos2MosCount
                        begin
                        Col = DataGridViewColumn.Create( "Pct" +
                         NumToStr( M2MBack, 0 ) + "P" ) ;
                        Col.Width = 50 ;

                        Grd.columns.Add( Col ) ;
                        end ;

                for M2MBack = 1 TO iMos2MosCount
                        begin
                        Col = DataGridViewColumn.Create( "Avg" +
                         NumToStr( M2MBack, 0 ) + "P" ) ;
                        Col.Width = 50 ;
                        Grd.columns.Add( Col ) ;
                        end ;

                { show form }
                FGrid.Show() ;

        catch ( Exception ex )
                Print( Name, " AnalysisTechnique_Initialized - ErrMsg= ", ex.Message
) ;
        end ;
        end ;

method void SortIntoVector()
variables:
        int VectorIndex,
        int ToAddCount,
        Dictionary VectorPatternData,
        bool Added ;

        begin
        SortVector.clear();
        for PatternIndex = 0 TO Patterns.Count - 1
                begin
                PatternData = Patterns.Values[PatternIndex] astype Dictionary ;
```

```
                ToAddCount = PatternData["Count"] astype int ;
                Added = false;

                for VectorIndex = 0 TO SortVector.Count - 1
                        begin
                        VectorPatternData = SortVector[VectorIndex] astype Dictionary
;

                        if VectorPatternData["Count"] astype int < ToAddCount then
                                begin
                                SortVector.insert( VectorIndex, PatternData ) ;
                                Added = true ;
                                break ;
                                end ;


                        end ;
                { add to end of vector if not added }
                if not Added then
                        begin
                        SortVector.push_back( PatternData ) ;
                        end ;
                end ;

        end ;

{ main Loop through bars }
try
        { only process after a number of bars to cover M2M size }
        if BarStatus( DataNum + 1 ) = bsClosing then
                begin

                AverageRange = Average(Range, iAvgRngLen);
                BarData = Vector.Create();

                { We are using the vector like a class, with specific values stored
at
                  specific  indices as defined by the bv... constants }
                for Value1 = 0 to bvLast
                        begin
                        BarData.push_back( 0 ) ;
                        end ;

                BarData[bvNumber] = BarNumber ;
                BarData[bvOpen] = Open ;
                BarData[bvHigh] = High ;
                BarData[bvLow] = Low ;
                BarData[bvClose] = Close ;
                Bardata[bvAvgRange] = AverageRange ;

                { categorize bar pattern }
                CandleRange = Range ;
                { force to 1 if > 1 }
                RangeMultiplier = MinList( 1, CandleRange / AverageRange ) ;
                CandleRange = CandleRange / 100.0 ;

                HO = Round( ( ( ( High - Open ) / CandleRange ) * RangeMultiplier ) /
                 SegmentDivisor, 0 ) ;
                HC = Round( ( ( ( High - Close ) / CandleRange ) * RangeMultiplier )
/
```

```
                 SegmentDivisor, 0 ) ;
                OL = Round( ( ( ( Open - Low ) / CandleRange ) * RangeMultiplier ) /
                 SegmentDivisor, 0 ) ;

                CandleSignature = NumToStr( HO, 0 ) + "-" + NumToStr( HC, 0 ) + "-" +
                 NumToStr( OL, 0 ) ;
                Bardata[bvSignature] = CandleSignature ;
                BarsData.Add( BarNumber.tostring(), BarData ) ;

                { create patterns dictionary entry }
                if NOT Patterns.Contains( CandleSignature ) then
                        begin
                        PatternData = Dictionary.Create() ;
                        PatternData.Add( "Signature", CandleSignature ) ;
                        PatternData.Add( "Count", 0 ) ;

                        for M2MBack = 1 TO iMos2MosCount
                                begin
                                PatternData.Add( "PctCount" + NumToStr( M2MBack, 0 ),
0 ) ;
                                PatternData.Add( "PctUpCount" + NumToStr( M2MBack, 0
), 0 ) ;
                                PatternData.Add( "AvgCumm" + NumToStr( M2MBack, 0 ),
0.0 ) ;
                                end ;

                        Patterns.Add( CandleSignature, PatternData ) ;
                        PatternsCount = Patterns.Count ;
                        end ;

                { record that this pattern was found in a bar }
                PatternData = Patterns[CandleSignature] astype Dictionary ;
                PatternData["Count"] = PatternData["Count"] astype int + 1 ;

                {
                As each new bar is encountered, this code accumulates the month-
to-month
                data for each of the patterns in the prior bars convered by the
                iMos2MosCount parameter.  We must ensure that we do not step back
before
                first bar
                }
                for M2MBack = 1 To iMos2MosCount
                        begin
                        M2MBackStr = NumToStr(M2MBack,0);

                        { Only process backwards to BarNumber 1 }
                        if CurrentBar - M2MBack >= 1 then
                                begin

                                { Get BarData for M2MBack from current bar }
                                PatternBarNo = CurrentBar - M2MBack ;
                                PatternBarNoStr = NumToStr( PatternBarNo, 0 ) ;
                                PatternBarData = BarsData[PatternBarNoStr] astype
Vector ;

                                { Get Patterns PatternData Dictionary }
                                PatternData = Patterns[PatternBarData[bvSignature]
astype string]
                                 astype Dictionary ;
```

```
                                {       Update PctCount, PctUpCount, AvgCumm }
                                PatternData["PctCount" + M2MBackStr] =
                                 PatternData["PctCount" + M2MBackStr] astype Int + 1 ;

                                if BarData[bvClose] astype Double >
PatternBarData[bvClose]
                                 astype Double then
                                        PatternData["PctUpCount" + M2MBackStr] =
                                         PatternData["PctUpCount" + M2MBackStr]
astype Int + 1 ;

                                Patterndata["AvgCumm" + M2MBackStr] =
Patterndata["AvgCumm" +
                                 M2MBackStr] astype Double + ( BarData[bvClose]
astype Double -
                                 PatternBarData[bvClose] astype Double ) /
                                 PatternBarData[bvClose] astype Double ;

                                end ;
                        end ;

                end ;
catch ( Exception ex )
        print( Name, " MAIN - Error=", ex.Message ) ;
        print( Name, " MAIN - Source=", ex.Source ) ;
        print( Name, " MAIN - Target=", ex.TargetSite ) ;
end;

#region - Display Data in Grid in form -

if LastBarOnChartEx then
        begin
        try
                Grd.Rows.Clear() ;
                SortIntoVector() ;

                for PatternIndex = 0 to SortVector.Count - 1
                        begin
                        Row = DataGridViewRow.Create("");
                        Grd.Rows.Add(Row);

                        PatternData = SortVector[PatternIndex] astype Dictionary;
                        Row.Cells[0].Text = PatternData["Signature"] astype string;

                        Row.Cells[1].Text = (PatternData["Count"] astype
Int).ToString();

                        for M2MBack = 1 to iMos2MosCount
                                begin
                                { Calculate Pct<n>P = WinCount / Count }
                                Value1 = PatternData["PctUpCount" + NumToStr(
M2MBack, 0 )]
                                 astype int ;
                                Value2 = PatternData["PctCount" + NumToStr( M2MBack,
0)]
                                 astype int ;
                                Row.Cells[1 + M2MBack].Text =
                                 NumToStr( ( Value1 / Value2 ) * 100, 0 ) + "%"
;
```

```
                                end ;

                        for M2MBack = 1 to iMos2MosCount
                                begin
                                Value1 = PatternData["AvgCumm" + NumToStr( M2MBack, 0
)]
                                 astype Double ;
                                Row.Cells[1 + iMos2MosCount + M2MBack].Text =
                                 NumToStr( Round( Value1 * 100 / Value2, 2 ), 2 ) ;
                                end ;

                        end ;

        catch ( Exception ex )
                print( Name, " MAIN - Error=", ex.Message ) ;
                print( Name, " MAIN - Source=", ex.Source ) ;
                print( Name, " MAIN - Target=", ex.TargetSite ) ;
        end ;

        #endregion

        end ;
_CondensedCandlestickLocator (Indicator)

inputs:
        TargetHO( 4 ),
        TargetHC( 1 ),
        TargetOL( 1 ),
        AvgRangeLength( 10 ) ;

variables:
        SegmentDivisor( 0 ),
        AverageRange( 0 ),
        CandleRange( 0 ),
        RangeMultiplier( 0 ),
        CandleRangeSegmented( 0 ),
        HO( 0 ),
        HC( 0 ),
        OL( 0 ),
        Segments( 100 ),
        SegmentCount( 5 ) ;

{ signal calculation }
CandleRange = High - Low ;
SegmentDivisor = Segments/SegmentCount ;
AverageRange = Average( CandleRange, AvgRangeLength ) ;

if AverageRange[1] > 0 then
        begin
        RangeMultiplier = CandleRange/AverageRange[1] ;
        CandleRangeSegmented = CandleRange/Segments ;
        RangeMultiplier =  IFF( RangeMultiplier > 1, 1, RangeMultiplier ) ;

        HO = Round( ( ( ( High - Open ) / CandleRangeSegmented ) * RangeMultiplier )
/
         SegmentDivisor, 0 ) ;
        HC = Round( ( ( ( High - Close ) / CandleRangeSegmented ) * RangeMultiplier )
/
         SegmentDivisor, 0 ) ;
        OL = Round( ( ( ( Open - Low ) / CandleRangeSegmented ) * RangeMultiplier ) /
```

```
 SegmentDivisor, 0 ) ;

if HO = TargetHO and HC = TargetHC and OL = TargetOL then
        Plot1( Close ) ;
end ;
```

— Christopher Davy and Mark Mills
TradeStation Securities, Inc.
www.TradeStation.com

BACK TO LIST

eSIGNAL: FEBRUARY 2015

For this month's Traders' Tip, we've provided a CondensedCandles.efs study based on the formula described in Dave Cline's article in this issue, "Candlesticks, Condensed."

The data generated by the formula will be found in the Candle.csv file located in the My Documents/Interactive Data/FormulaOutput folder. This data can then be used to do the probability analysis based on candle patterns described in the article.

The study contains formula parameters that may be configured through the *edit chart* window (right-click on the chart and select "edit chart"). A sample chart is shown in Figure 2.

	A	B	C	D	E	F	G	H
1	Signature	Rank	Count	Ups	Downs	Total	AvgReturn	PctUp
2	-5:1:1	-3.24	9	2	7	-65.52	-7.28	0.222
3	+0:3:6	-3.16	3	2	1	-7.1	-2.3666667	0.667
4	+0:2:6	-2.34	2	1	1	-9.38	-4.69	0.5
5	-3:3:2	-2.27	2	1	1	-9.08	-4.54	0.5
6	+1:3:5	-1.86	3	2	1	-4.184	-1.39479	0.667
7	-2:1:2	-1.84	3	2	1	-4.13	-1.3766667	0.667
8	-2:2:2	-1.6	3	1	2	-14.39	-4.79604	0.333
9	-3:1:1	-1.21	4	1	3	-19.3	-4.825	0.25
10	+1:5:5	-1.06	4	1	3	-16.9	-4.22375	0.25

FIGURE 2: eSIGNAL. Here is an example of the study implemented on a chart of XLE (Energy Select Sector SPDR).

To discuss this study or download a complete copy of the formula code, please visit the EFS Library Discussion Board forum under the *forums* link from the support menu at www.esignal.com or visit our EFS KnowledgeBase at http://www.esignal.com/support/kb/efs/. The eSignal formula script (EFS) is also available for downloading here and copying & pasting from below.

```
/*********************************
Provided By:
    Interactive Data Corporation (Copyright © 2014)
    All rights reserved. This sample eSignal Formula Script (EFS)
    is for educational purposes only. Interactive Data Corporation
    reserves the right to modify and overwrite this EFS file with
    each new release.

Description:
    Candlesticks, Condensed by David Cline

Formula Parameters:                     Default:
Weight Periods                          50
Segment Count                           20
Report Mode                             Include all candles
```

```
Version:            1.00  12/08/2014

Notes:
    The related article is copyrighted material. If you are not a subscriber
    of Stocks & Commodities, please visit www.traders.com.

**********************************/

var fpArray = new Array();

function preMain(){

    setStudyTitle("CondensedCandlesticks");
    setPriceStudy(true);

    setCursorLabelName("Bar Signature", 0);
    setComputeOnClose(true);

    var x = 0;

    fpArray[x] = new FunctionParameter("fpPeriods", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("Weight Periods");
        setLowerLimit(1);
        setDefault(10);
    }

    fpArray[x] = new FunctionParameter("fpSegCount", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("Segment Count");
        setLowerLimit(1);
        setDefault(6);
    }

    fpArray[x] = new FunctionParameter("fpRepMode", FunctionParameter.STRING);
    with(fpArray[x++]){
        setName("Report Mode");
        addOption("Include all candles");
        addOption("Include the top and bottom 10 candles");
        setDefault("Include all candles");
    }
}

var bInit = false;
var bVersion = null;

var xOpen = null;
var xHigh = null;
var xLow = null;
var xClose  = null;

var candlePatterns = {};
var nSegmentDivisor = null;

var fCandles = new File("Candles.csv");

//The Script outputs the results of the analysis of Candles signatures
//to the file "C:\Users\User_name\Documents\Interactive Data\FormulaOutput
\Candles.csv"
```

```
function main(fpPeriods, fpSegCount, fpRepMode){

    if (bVersion == null) bVersion = verify();
    if (bVersion == false) return;

    nSegmentDivisor = 100.0 / fpSegCount;

    if (!bInit){

        xOpen = open();
        xHigh = high();
        xLow = low();
        xClose = close();

        xRange = efsInternal('calc_Range', xHigh, xLow);
        xAverageRange = sma(fpPeriods, xRange);

        bInit = true;
    }

    if (getBarState() == BARSTATE_ALLBARS){

        candlePatterns = {};

        fCandles.open("wt");
        fCandles.writeln("Signature, Rank, Count, Ups, Downs, Total, AvgReturn,
PctUp");
        fCandles.close();
    }

    var nCandleRange = xRange.getValue(-1);
    var nAverageRange = xAverageRange.getValue(0);

    if (nAverageRange == null || nCandleRange == null)
        return;

    var nRangeMultiplier = nCandleRange / nAverageRange;
    if (nRangeMultiplier > 1) nRangeMultiplier = 1;
    var nCandleRange = nCandleRange / 100;

    var nPrOpen = xOpen.getValue(-1);
    var nPrHigh = xHigh.getValue(-1);
    var nPrLow = xLow.getValue(-1);
    var nPrClose = xClose.getValue(-1);

    var nClose = xClose.getValue(0);
    var nStartClose = xClose.getValue(-fpPeriods + 1);

    if (nPrOpen == null || nPrHigh == null ||
        nPrLow == null || nPrClose == null ||
        nStartClose == null)
        return;

    var nHO = Math.round((((nPrHigh - nPrOpen) / nCandleRange) * nRangeMultiplier ) /
nSegmentDivisor);
    var nHC = Math.round((((nPrHigh - nPrClose) / nCandleRange) * nRangeMultiplier )
/ nSegmentDivisor);
    var nOL = Math.round((((nPrOpen - nPrLow) / nCandleRange) * nRangeMultiplier ) /
nSegmentDivisor);
```

```
    var stSign = null;

    if (nPrOpen > nClose) stSign = '+'
    else stSign = '-';

    var stCandleSignature = stSign + nHO + ':' + nHC + ':' + nOL;

    var nCandleReturn = nStartClose - nClose;

    if  (candlePatterns.hasOwnProperty(stCandleSignature)){

        if (nCandleReturn > 0) candlePatterns[stCandleSignature]['Ups'] += 1
        else candlePatterns[stCandleSignature]['Ups'] += 0;

        if (nCandleReturn <= 0) candlePatterns[stCandleSignature]['Downs'] += 1
        else candlePatterns[stCandleSignature]['Downs'] += 0;

        candlePatterns[stCandleSignature]['Total'] += nCandleReturn;
        candlePatterns[stCandleSignature]['Count'] += 1;
    }
    else{

        candlePatterns[stCandleSignature] = {};

        if (nCandleReturn > 0) candlePatterns[stCandleSignature]['Ups'] = 1
        else candlePatterns[stCandleSignature]['Ups'] = 0;

        if (nCandleReturn <= 0) candlePatterns[stCandleSignature]['Downs'] = 1
        else candlePatterns[stCandleSignature]['Downs'] = 0;

        candlePatterns[stCandleSignature]['Total'] = nCandleReturn;
        candlePatterns[stCandleSignature]['Count'] = 1;
    }

    if (getCurrentBarIndex() == -1)
        processCandleReturns(fpRepMode);

    return stCandleSignature;
}

function calc_Range(xHigh, xLow){

    return xHigh.getValue(0) - xLow.getValue(0);
}

function processCandleReturns(stRepMode){

    var arrCandlePatterns = [];

    for (candle in candlePatterns){
        candleMetric = candlePatterns[candle];
        candleMetric['PctUp'] = candleMetric['Ups'] / candleMetric['Count'];
        candleMetric['AvgReturn'] = candleMetric['Total'] / candleMetric['Count'];
        candleMetric['Rank'] = candleMetric['PctUp'] * candleMetric['Ups'] *
candleMetric['AvgReturn'];

        arrCandlePatterns.push([candle, candlePatterns[candle]]);
    }

    arrCandlePatterns.sort(function(a, b){
```

```
        return a[1]['Rank'] - b[1]['Rank'];
    });

    fCandles.open("at");
    if (stRepMode == "Include the top and bottom 10 candles" &&
arrCandlePatterns.length > 20){
        for (i = 0; i < 10; i++){
            fCandles.writeln(arrCandlePatterns[i][0] + ", " +
                             arrCandlePatterns[i][1]['Rank'] + ", " +
                             arrCandlePatterns[i][1]['Count'] + ", " +
                             arrCandlePatterns[i][1]['Ups'] + ", " +
                             arrCandlePatterns[i][1]['Downs'] + ", " +
                             arrCandlePatterns[i][1]['Total'] + ", " +
                             arrCandlePatterns[i][1]['AvgReturn'] + ", " +
                             arrCandlePatterns[i][1]['PctUp']);
        }
        for (i = arrCandlePatterns.length - 10; i < arrCandlePatterns.length; i++){
            fCandles.writeln(arrCandlePatterns[i][0] + ", " +
                             arrCandlePatterns[i][1]['Rank'] + ", " +
                             arrCandlePatterns[i][1]['Count'] + ", " +
                             arrCandlePatterns[i][1]['Ups'] + ", " +
                             arrCandlePatterns[i][1]['Downs'] + ", " +
                             arrCandlePatterns[i][1]['Total'] + ", " +
                             arrCandlePatterns[i][1]['AvgReturn'] + ", " +
                             arrCandlePatterns[i][1]['PctUp']);
        }
    }
    else
        for (i = 0; i < arrCandlePatterns.length; i++)
            fCandles.writeln(arrCandlePatterns[i][0] + ", " +
                             arrCandlePatterns[i][1]['Rank'] + ", " +
                             arrCandlePatterns[i][1]['Count'] + ", " +
                             arrCandlePatterns[i][1]['Ups'] + ", " +
                             arrCandlePatterns[i][1]['Downs'] + ", " +
                             arrCandlePatterns[i][1]['Total'] + ", " +
                             arrCandlePatterns[i][1]['AvgReturn'] + ", " +
                             arrCandlePatterns[i][1]['PctUp']);
    fCandles.close();
}

function verify(){

    var b = false;
    if (getBuildNumber() < 779){

        drawTextAbsolute(5, 35, "This study requires version 8.0 or later.",
            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,
            null, 13, "error");
        drawTextAbsolute(5, 20, "Click HERE to upgrade.@URL=http://www.esignal.com
/download/default.asp",
            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,
            null, 13, "upgrade");
        return b;
    }
    else
        b = true;

    return b;
```

```
}
```

—Eric Lippert
eSignal, an Interactive Data company
800 779-6555, www.eSignal.com

BACK TO LIST

WEALTH-LAB: FEBRUARY 2015

To seasoned readers, the article in this issue by Dave Cline, “Candlesticks, Condensed,” may remind them of a loosely related idea featured in the March 2001 and September 2001 issues of STOCKS & COMMODITIES by Viktor Likhovidov, who also offered a candlestick coding technique. His quantitative approach, which he called *CandleCode*, expresses a candlestick value as a binary number, coding the body and shadows in different “bits” of the number.

The idea behind the “candlesticktistics” introduced in Cline’s article in this issue is to make candlesticks a sort of binary, simplified, consistent instrument with which to identify patterns. By turning them into a fixed number of patterns, the author’s automatic pattern-search method performs statistical analysis in order to gauge their predictive capability. Our preliminary testing suggests that it delivers quite interesting results.

The C# code we’re presenting this month based on Cline’s article splits the backtest range into two equal parts: in-sample and out-of-sample periods. At first, it discovers the most popular HO:HC:OL signatures across the in-sample range. Leaving the actual candlestick probability programming to motivated users, we enter quick trades using the knowledge gained on in-sample data. The number of signatures is configurable; the more the Top-N consists of, the more trades will there be and vice versa. For example, if you choose to trade only the Top-1 pattern, the resulting system will be very selective. The percentage of price change following a pattern is also interactively configurable through a parameter slider at the bottom of the screen. By default, each trade lasts for a single bar only and is exited at the close of the entry bar.

On a closing note, our code is powered by LINQ, which is a C# / .NET way of producing flexible SQL-like statements on any data contained in arrays, lists, databases and so on. Advanced users can leverage the benefits of LINQ to perform statistical analysis on candlestick patterns in terse, compact, and almost natural language form.

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Drawing;
using WealthLab;
using WealthLab.Indicators;
using System.Linq;

namespace WealthLab.Strategies
```

```
{
        public class Signature
        {
                public int Bar;
                public string Sig;
                public double ROC;

                public Signature( int Bar, string Sig, double ROC ){ this.Bar = Bar;
this.Sig = Sig; this.ROC = ROC;}
        }

        public class TASC201502 : WealthScript
        {
                private StrategyParameter paramTop;
                private StrategyParameter paramRoc;

                public TASC201502()
                {
                        paramTop = CreateParameter("Top N",10,1,10,1);
                        paramRoc = CreateParameter("ROC period",1,1,10,1);
                }

                protected override void Execute()
                {
                        int roc = paramRoc.ValueInt;
                        const char tab = '\u0009';
                        int halfData = Bars.Count / 2; // For in-sample and out-of-
sample
                        int oosBar = Bars.Count - halfData;
                        int segmentCount = 5;
                        double segmentDivisor = 100d / (double)segmentCount;
                        List<Signature> lstSig1 = new
List<Signature>();

                        if (Bars.Count < roc )
                        {
                                DrawLabel(PricePane, "Insufficient data");
                                Abort();
                        }

                        for(int bar = GetTradingLoopStartBar(10); bar < Bars.Count;
bar++)
                        {
                                List<string> lstSig = new List<string>();
                                double open = Open[bar], high = High[bar], low =
Low[bar], close = Close[bar];

                                double averageRange = ATR.Series(Bars,10)[bar];
                                double candleRange = high - low;
                                double rangeMultiplier = candleRange / averageRange;
                                candleRange = candleRange /= 100d;
                                rangeMultiplier = rangeMultiplier > 1 ? 1.0 :
rangeMultiplier;

                                double HO = Math.Round((((high - open) / candleRange)
* rangeMultiplier) / segmentDivisor);
                                double HC = Math.Round((((high - close) /
candleRange) * rangeMultiplier) / segmentDivisor);
                                double OL = Math.Round((((open - low) / candleRange)
* rangeMultiplier) / segmentDivisor);
```

```
                                //Candle signature = HO-HC-OL
                                var sig = HO+":"+HC+":"+OL;
                                if( bar < oosBar )
                                        lstSig1.Add(new
Signature(bar,sig,ROC.Series(Close,roc)[bar+1]));
                                else
                                {
                                        if (IsLastPositionActive)
                                        {
                                                SellAtClose(bar,LastPosition);
                                        }
                                        else
                                        {
                                                var listSignaturesOOS =
lstSig1.OrderBy(p => p.ROC).Reverse().Take(paramTop.ValueInt).ToList();
                                                foreach(var s in listSignaturesOOS )
                                                {
                                                        if( sig == s.Sig )
                                                        {
                                                                int count =
lstSig1.Count( t => t.Sig == sig );

AnnotateBar(s.Sig,bar,true,Color.Blue);
                                                                BuyAtClose(bar);
                                                                break;
                                                        }
                                                }
                                        }
                                }
                        }

                        var listSignaturesIS = lstSig1.OrderBy(p =>
p.ROC).Reverse().Take(paramTop.ValueInt).ToList();

                        DrawLabel(PricePane, "Most frequent signatures from in-sample
period containing 50% data: " + "\n\r");
                        DrawLabel(PricePane, "Signature:" + tab + "Count:" + tab + "%
Change next bar:");
                        foreach(var s in listSignaturesIS )
                        {
                                int count = lstSig1.Count( t => t.Sig == s.Sig );
                                DrawLabel(PricePane, s.Sig + tab + tab + count + tab
+ Bars.FormatValue(s.ROC));
                                AnnotateBar(s.Sig,s.Bar,true,Color.Black);
                        }

                        AnnotateBar("Out of sample starts",oosBar,true,Color.Red);
                        for( int bar = 0; bar < oosBar; bar++ )
                                SetBackgroundColor(bar,Color.FromArgb(30,Color.Blue));
                }
        }
}
```

A sample chart is shown in Figure 3.

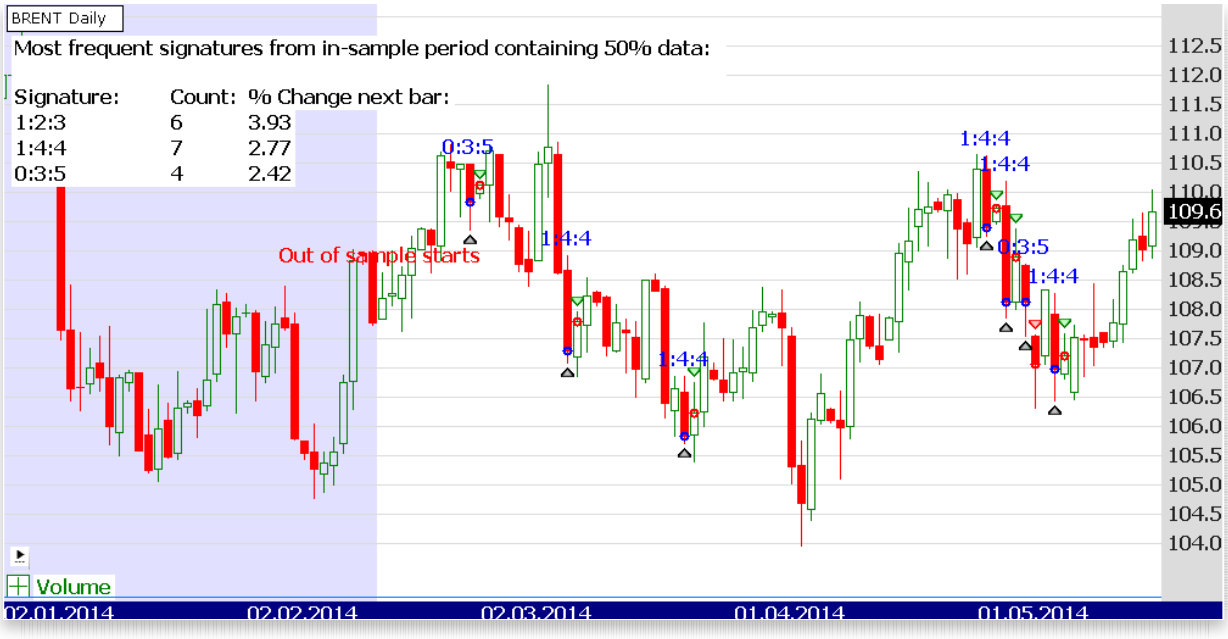


FIGURE 3: WEALTH-LAB. Here is a sample Wealth-Lab 6 chart illustrating the application of top-3 candlestick patterns on out-of-sample data using a daily chart of a continuous contract of crude oil futures (brent).

—Eugene, Wealth-Lab team
MS123, LLC
www.wealth-lab.com

BACK TO LIST

AMIBROKER: FEBRUARY 2015

In "Candlesticks, Condensed" in this issue, author Dave Cline presents a way to condense candlestick chart information so that statistics about candlestick patterns can be derived. The AmiBroker code provided here presents an AmiBroker Formula Language (AFL) formula that produces condensed candlestick pattern statistics similar to those presented in Cline's article.

LISTING 1.

```
SegmentCount = 5;
SegmentDivisor = 100 / SegmentCount;
CandleRange = High - Low;
AvgRange = MA( CandleRange, 6 );
RangeMultiplier = CandleRange / AvgRange;
CandleRange /= 100;

RangeMultiplier = Min( RangeMultiplier, 1 );
```

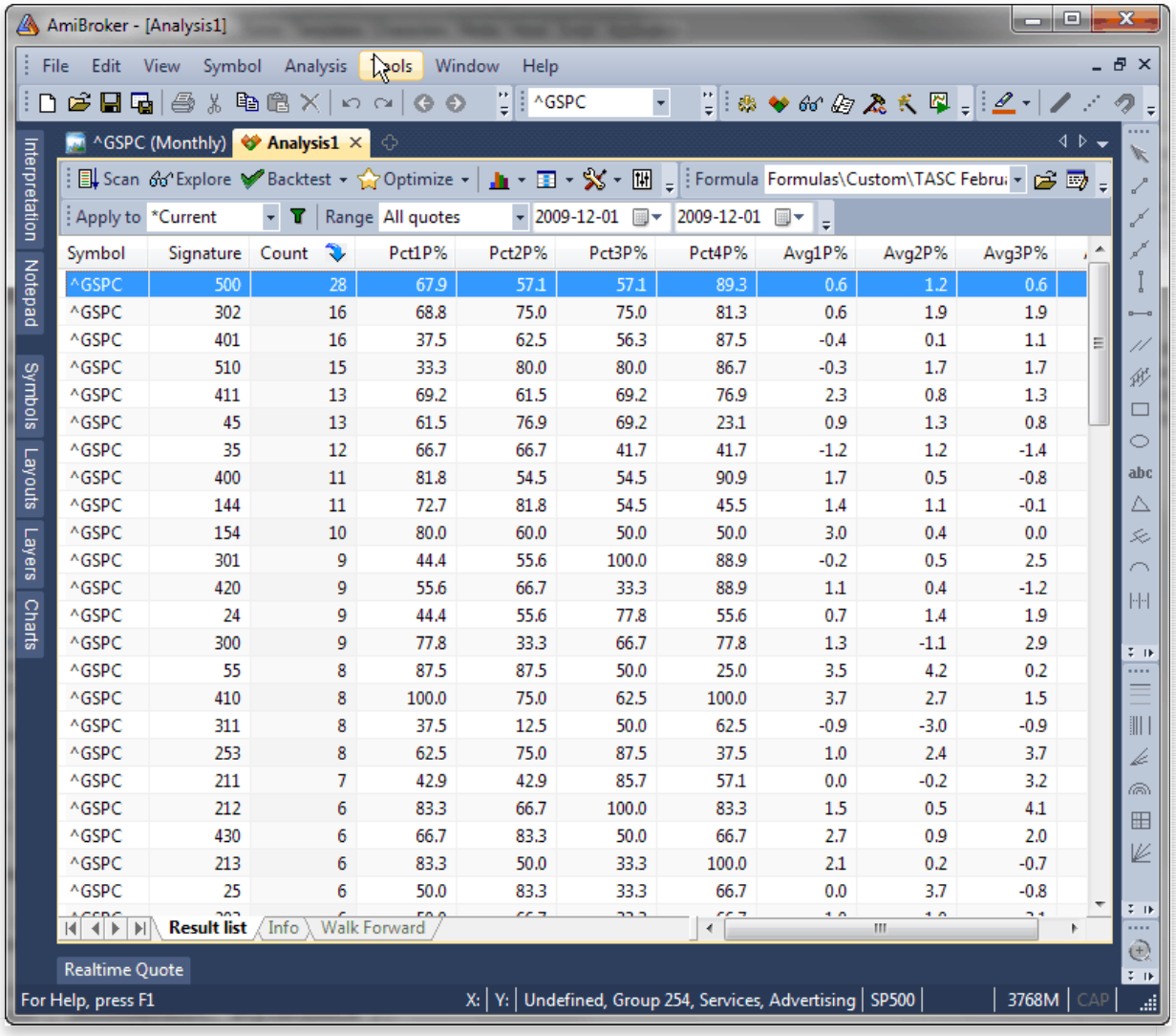
```
HO = round( ( RangeMultiplier * ( High - Open ) / CandleRange ) / SegmentDivisor );
HC = round( ( RangeMultiplier * ( High - Close ) / CandleRange ) / SegmentDivisor );
OL = round( ( RangeMultiplier * ( Open - Low ) / CandleRange ) / SegmentDivisor );

DigitMult = SegmentCount + 1;

// signature is encoded to fit into integer
CandleSignature = DigitMult * DigitMult * HO + DigitMult * HC + OL;

Title = Name() + " " + Date() + " " + StrFormat( "Signature %.0f:%.0f:%.0f", HO, HC,
OL );
Plot( C, "Close", colorDefault, styleCandle );

Filter = False;

// statistics
Pct1P = Pct2P = Pct3P = Pct4P = 0;
Avg1P = Avg2P = Avg3P = Avg4P = 0;

// count of occurrence of patterns
Cnt = 0;

MaxSegment = SegmentCount * DigitMult * DigitMult + SegmentCount * DigitMult +
SegmentCount;

if( BarCount > MaxSegment )
{
   // changes over 1..4 periods
   // you can change it to Close - Ref( Close, -1 )
   // to get dollar gains instead of percent changes

   Chg1P = ROC( Close, 1 );
   Chg2P = ROC( Close, 2 );
   Chg3P = ROC( Close, 3 );
   Chg4P = ROC( Close, 4 );

   // number of times given pattern is followed by UP movement
   Up1P = Up2P = Up3P = Up4P = 0;
   // gain
   Gain1P = Gain2P = Gain3P = Gain4P = 0;

   for( i = 0; i < BarCount - 4; i++ )
   {
       sig = CandleSignature[ i ];

       if( sig >= 0 && sig <= MaxSegment )
       {
         Cnt[ sig ]++; // increase occurrence counter for given sig

         // if change > 0 increase UP counter
           Up1P[ sig ] += Chg1P[ i + 1 ] > 0;
           // add change to total gain
         Gain1P[ sig ] += Chg1P[ i + 1 ];

         Up2P[ sig ] += Chg1P[ i + 2 ] > 0;
         Gain2P[ sig ] += Chg1P[ i + 2 ];

         Up3P[ sig ] += Chg1P[ i + 3 ] > 0;
         Gain3P[ sig ] += Chg1P[ i + 3 ];
```

```
        Up4P[ sig ] += Chg4P[ i + 1 ] > 0;
        Gain4P[ sig ] += Chg4P[ i + 1 ];
      }
    }


    for( sig = 0; sig <= MaxSegment; sig++ )
    {
       // if any patterns of given signature were found
      if( Cnt[ sig ] )
      {
          qty = Cnt[ sig ];
          Pct1P[ sig ] = Up1P[ sig ] / qty;
          Pct2P[ sig ] = Up2P[ sig ] / qty;
          Pct3P[ sig ] = Up3P[ sig ] / qty;
          Pct4P[ sig ] = Up4P[ sig ] / qty;

          Avg1P[ sig ] = Gain1P[ sig ] / qty;
          Avg2P[ sig ] = Gain2P[ sig ] / qty;
          Avg3P[ sig ] = Gain3P[ sig ] / qty;
          Avg4P[ sig ] = Gain4P[ sig ] / qty;
       }
    }

    Filter = Cnt;
}
else
{
    PopupWindow("Not enough bars in " + Name() + " Please select symbols that have
more than " +
                MaxSegment + " bars", "Problem" );
}

// multiply factors by 100 to get percents
Pct1P *= 100;
Pct2P *= 100;
Pct3P *= 100;
Pct4P *= 100;

bi = BarIndex();
Sig1 = bi % DigitMult;
Sig2 = floor( bi / DigitMult ) % DigitMult;
Sig3 = floor( bi / ( DigitMult * DigitMult ) );

Signature = 100 * Sig3 + 10 * Sig2 + Sig1;

SetOption("NoDefaultColumns", True );
AddTextColumn( Name(), "Symbol" );
AddColumn( Signature, "Signature", 3.0 );
AddColumn( Cnt, "Count", 1.0 );
AddColumn( Pct1P, "Pct1P%", 1.1 );
AddColumn( Pct2P, "Pct2P%", 1.1 );
AddColumn( Pct3P, "Pct3P%", 1.1 );
AddColumn( Pct4P, "Pct4P%", 1.1 );
AddColumn( Avg1P, "Avg1P%", 1.1 );
AddColumn( Avg2P, "Avg2P%", 1.1 );
AddColumn( Avg3P, "Avg3P%", 1.1 );
AddColumn( Avg4P, "Avg4P%", 1.1 );
```

When used in AmiBroker's Exploration mode, it produces a table showing the number of occurrences of a given pattern along with its profitability within the subsequent one to four periods. When used in indicator mode, it shows the current pattern code in the title string of the chart.

To use the formula, enter the code into the formula editor and press the *apply indicator* button to use it as a chart or the send to analysis button to perform an exploration.

A sample chart is shown in Figure 4.



Symbol	Signature	Count	Pct1P%	Pct2P%	Pct3P%	Pct4P%	Avg1P%	Avg2P%	Avg3P%
^GSPC	500	28	67.9	57.1	57.1	89.3	0.6	1.2	0.6
^GSPC	302	16	68.8	75.0	75.0	81.3	0.6	1.9	1.9
^GSPC	401	16	37.5	62.5	56.3	87.5	-0.4	0.1	1.1
^GSPC	510	15	33.3	80.0	80.0	86.7	-0.3	1.7	1.7
^GSPC	411	13	69.2	61.5	69.2	76.9	2.3	0.8	1.3
^GSPC	45	13	61.5	76.9	69.2	23.1	0.9	1.3	0.8
^GSPC	35	12	66.7	66.7	41.7	41.7	-1.2	1.2	-1.4
^GSPC	400	11	81.8	54.5	54.5	90.9	1.7	0.5	-0.8
^GSPC	144	11	72.7	81.8	54.5	45.5	1.4	1.1	-0.1
^GSPC	154	10	80.0	60.0	50.0	50.0	3.0	0.4	0.0
^GSPC	301	9	44.4	55.6	100.0	88.9	-0.2	0.5	2.5
^GSPC	420	9	55.6	66.7	33.3	88.9	1.1	0.4	-1.2
^GSPC	24	9	44.4	55.6	77.8	55.6	0.7	1.4	1.9
^GSPC	300	9	77.8	33.3	66.7	77.8	1.3	-1.1	2.9
^GSPC	55	8	87.5	87.5	50.0	25.0	3.5	4.2	0.2
^GSPC	410	8	100.0	75.0	62.5	100.0	3.7	2.7	1.5
^GSPC	311	8	37.5	12.5	50.0	62.5	-0.9	-3.0	-0.9
^GSPC	253	8	62.5	75.0	87.5	37.5	1.0	2.4	3.7
^GSPC	211	7	42.9	42.9	85.7	57.1	0.0	-0.2	3.2
^GSPC	212	6	83.3	66.7	100.0	83.3	1.5	0.5	4.1
^GSPC	430	6	66.7	83.3	50.0	66.7	2.7	0.9	2.0
^GSPC	213	6	83.3	50.0	33.3	100.0	2.1	0.2	-0.7
^GSPC	25	6	50.0	83.3	33.3	66.7	0.0	3.7	-0.8

FIGURE 4: AMIBROKER. Here are sample exploration results showing the monthly pattern statistics for the S&P 500 index for the years 1990–2014.

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

BACK TO LIST

NEUROSHELL TRADER: FEBRUARY 2015

The technique described by Dave Cline in his article in this issue, “Candlesticks, Condensed” can be easily implemented with a few of NeuroShell Trader’s 800+ indicators plus the Advanced Indicator Set #2. Simply select *new indicator* from the *insert* menu and use the indicator wizard to recreate the following indicators:

```
RangeMultiplier: Min2(1,Avg Ratio (Sub(High,Low), 10))
HO: Round(Multiply3(Divide(Sub(High,Open),Sub(High,Low)), RangeMultiplier,
SegmentCount))
HC: Round(Multiply3(Divide(Sub(High,Close),Sub(High,Low)), RangeMultiplier,
SegmentCount))
HL: Round(Multiply3(Divide(Sub(Open,Low),Sub(High,Low)), RangeMultiplier,
SegmentCount))
Candle Signature: Add3( HO, HC, HL)
```

Neural networks can provide a more robust analysis of historical patterns than the segmented statistical analysis described in Cline’s article, so we chose to use the condensed candlesticks as inputs to a neural network. To create a neural network that analyzes and trades based on the past three condensed candlestick HO, HC and HL values, simply select *new prediction* from the *insert* menu and use the prediction wizard to create the following prediction inputs:

```
Non-rounded HO
Non-rounded HC
Non-rounded HL
Lag(Non-rounded HO,1)
Lag(Non-rounded HC,1)
Lag(Non-rounded HL,1)
Lag(Non-rounded HO,2)
Lag(Non-rounded HC,2)
Lag(Non-rounded HL,2)
```

Since neural network analysis also does not require grouping into integers as does the segmented statistical analysis, nonrounded versions of the indicators were used as inputs. Using six years of monthly bars for training and holding out an additional six years for out-of-sample evaluation, a quickly constructed nonoptimized prediction gave an average yearly return of 13.5% over the out-of-sample period for AAPL, ADBE, and ADP.

NeuroShell Trader also allows you to choose whether the parameters should be optimized. After backtesting the prediction, use the prediction analysis button to view the backtest and trade-by-trade statistics for the strategy.

Users of NeuroShell Trader can go to the STOCKS & COMMODITIES section of the NeuroShell Trader free technical support website to download a copy of this or any previous Traders’ Tips.

A sample chart is shown in Figure 5.

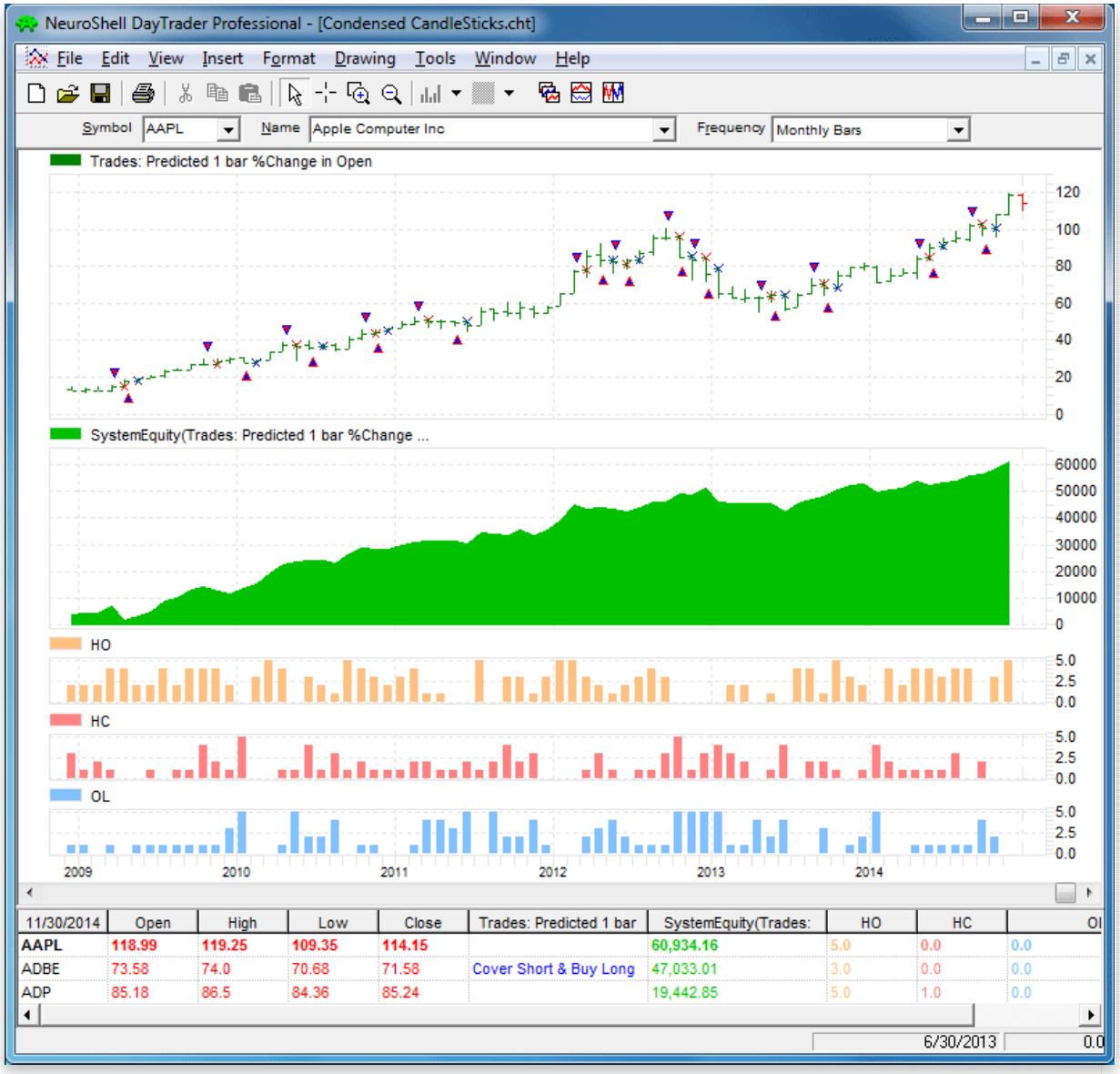


FIGURE 5: NEUROSHELL TRADER. Here is a NeuroShell Trader chart showing sample condensed candlestick neural network trading results.

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

BACK TO LIST

NINJATRADER®

NINJATRADER: FEBRUARY 2015

The CondensedCandles indicator introduced in “Candlesticks, Condensed” by Dave Cline in this issue has been made available for download at www.ninjatrader.com/SC/February2015SC.zip.

Once it has been downloaded, from within the NinjaTrader Control Center window, select the menu File → Utilities → Import NinjaScript and select the downloaded file. This file is for NinjaTrader version 7 or greater.

You can review the indicator source code by selecting the menu Tools → Edit NinjaScript → Indicator from within the NinjaTrader Control Center window and selecting the CondensedCandles file.

A sample chart implementation is shown in Figure 6.

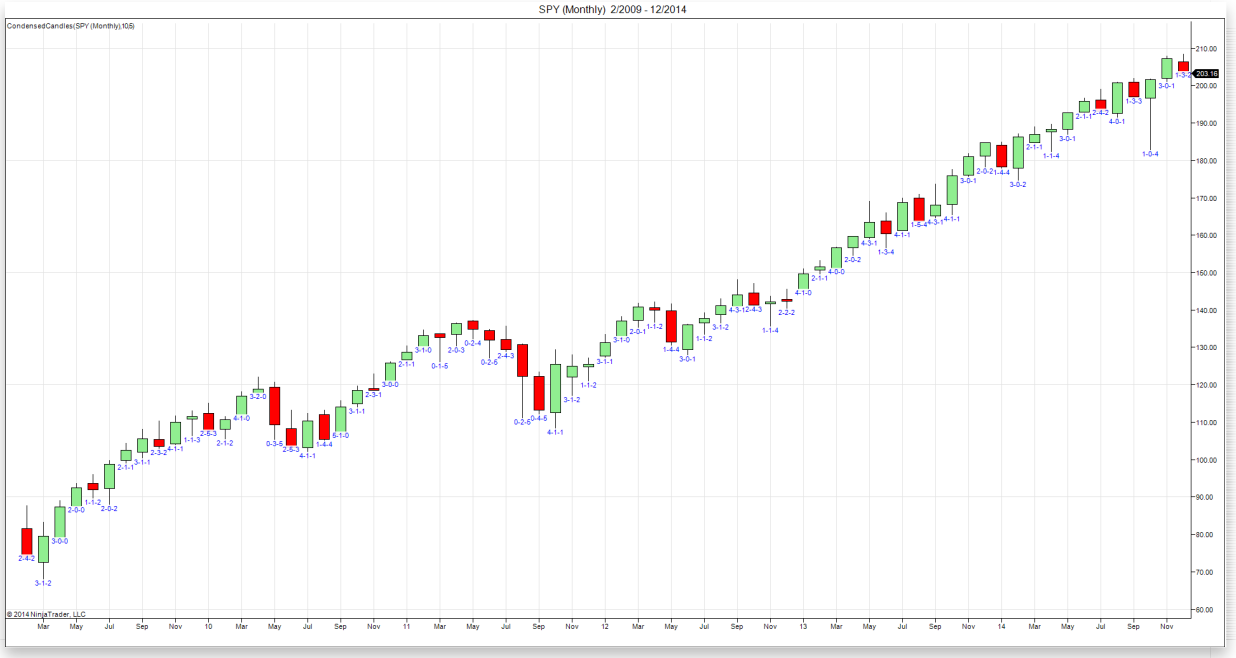


FIGURE 6: NINJATRADER. This screenshot shows the indicator applied to a monthly SPY ETF chart in NinjaTrader.

—Raymond Deux and Chelsea Bell
NinjaTrader, LLC, www.ninjatrader.com

BACK TO LIST

AIQ: FEBRUARY 2015

The AIQ code based on Dave Cline's article in this issue, "Candlesticks, Condensed," is

provided at www.TradersEdgeSystems.com/traderstips.htm, and shown below.

In preparing the coding in AIQ for the candle signatures, I used the weekly mode, since AIQ has a daily mode and a weekly mode but no monthly mode in the EDS module. I created a series of reports that show the candle signatures for the date entered into the report date box. The first report, named "ListAll," lists all stocks with closes greater than $5.00 in the database and the report shows the signature for the current week and also for the prior weekly bar. I also created reports for all of the signatures that are listed in the author's Figure 6. These additional reports all have the same format as the "ListAll" report. The "S411" report shows all stocks with prices greater than $5.00 that also have a current weekly signature of 4:1:1. These reports also list whatever the signature was for the prior week. Candle patterns can be constructed using multiple bars of signatures. For illustrative purposes only, I am providing the following format for a three-bar pattern:

```
S3_411 if S411 and valrule(S411,1) and valrule(S411,2)
```

This report lists only those stocks that had a 4:1:1 signature for the current bar and the prior two bars.

In Figure 7, I show a chart of eHealth (EHTH) as of January 10, 2014. On this date, it appeared on the S3_411 report. Buying on the next weekly open and selling at the following weekly open resulted in a 9.35% return.

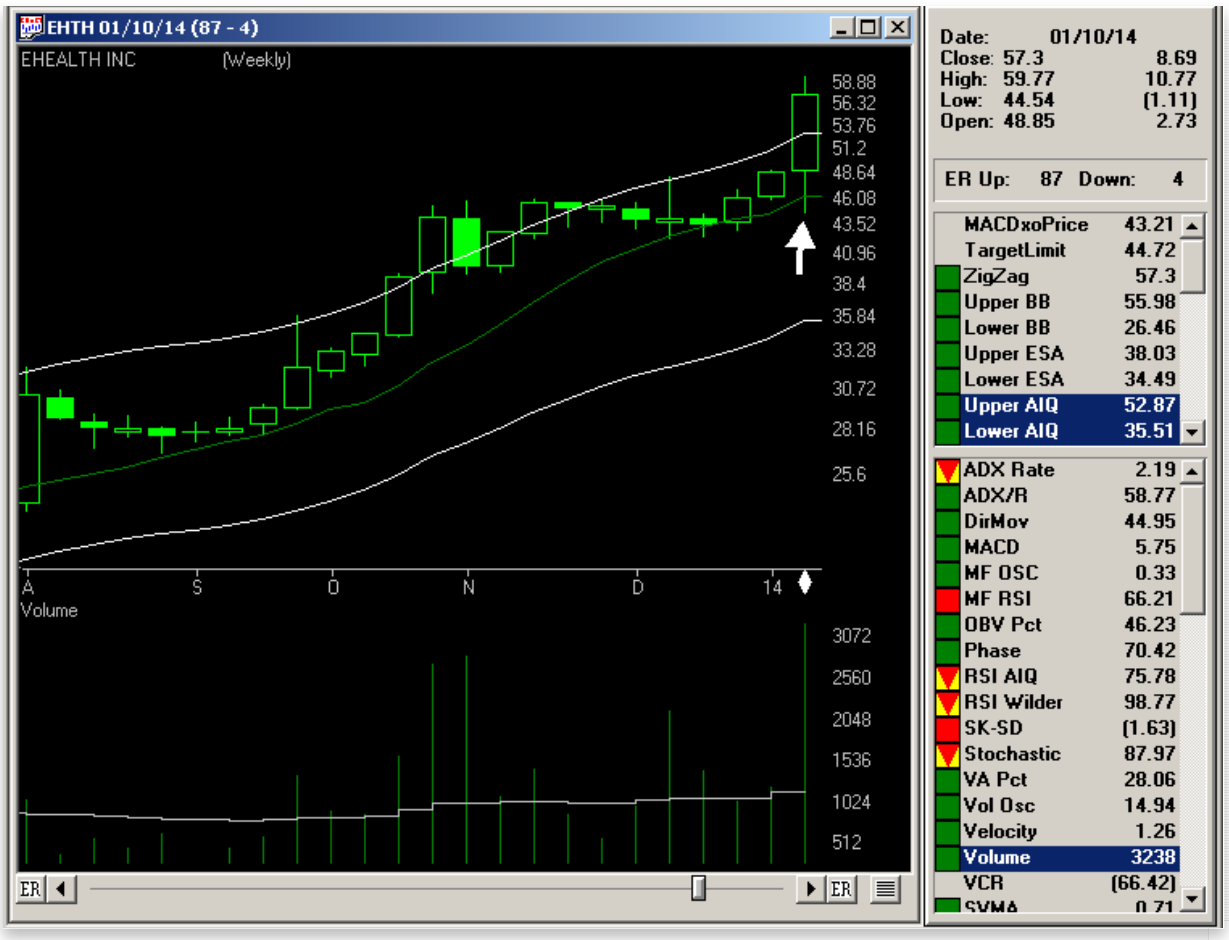


FIGURE 7: AIQ. Here is a weekly chart of eHealth (EHTH) as of 1/10/2014 showing white up arrows on the signal date for the S3_411 three-bar report pattern.

Note that I did not code exits for the patterns, as the built-in exits can be used to experiment with the candle signatures.

```
!CANDLESTICKS, CONDENSED
!Author: David Cline, TASC February 2015
!Coded by: Richard Denning, 12/10/2014
!www.TradersEdgeSystems.com

! NOTE: DESIGNED TO RUN IN WEEKLY MODE ONLY
!       SET PROPERTIES TO WEEKLY
!       RUN ON LAST DAY OF WEEK

! PARAMETERS:
AvgRngLen is 10.     !Length for range average
SegCt     is 5.      !Range = 3 to 6

! ABBREVIATIONS:
C is [close].
H is [high].
L is [low].
O is [open].
PD is {position days}.
PEP is {position entry price}.

! CONDENSED CANDLE CODE:
Range   is H - L.
AvgRng  is simpleavg(Range,AvgRngLen).
RngMult is Range / AvgRng.
SegDiv is 100 / SegCt.
CndlRng is Range / 100.
RngMult2 is iff(RngMult > 1,1,RngMult).
HO1 is H - O.
HC1 is H - C.
OL1 is O - L.
HO is round(((HO1/CndlRng)*RngMult2)/SegDiv).
HC is round(((HC1/CndlRng)*RngMult2)/SegDiv).
OL is round(((OL1/CndlRng)*RngMult2)/SegDiv).
HO_1 is valresult(HO,1).
HC_1 is valresult(HC,1).
OL_1 is valresult(OL,1).
HO_2 is valresult(HO,2).
HC_2 is valresult(HC,2).
OL_2 is valresult(OL,2).

!REPORTS:
ListAll if C>5.
S411 if HO = 4 and HC = 1 and OL = 1 and C>5.
S144 if HO = 1 and HC = 4 and OL = 4 and C>5.
S124 if HO = 1 and HC = 2 and OL = 4 and C>5.
S510 if HO = 5 and HC = 1 and OL = 0 and C>5.
S302 if HO = 3 and HC = 0 and OL = 2 and C>5.
```

```
S025 if HO = 0 and HC = 2 and OL = 5 and C>5.
S035 if HO = 0 and HC = 3 and OL = 5 and C>5.
S421 if HO = 4 and HC = 2 and OL = 1 and C>5.
S410 if HO = 4 and HC = 1 and OL = 0 and C>5.
S312 if HO = 3 and HC = 1 and OL = 2 and C>5.
!Example of how to construct a three bar pattern:
S3_411 if S411 and valrule(S411,1) and valrule(S411,2).
```

—Richard Denning
info@TradersEdgeSystems.com
for AIQ Systems

BACK TO LIST

TRADERSSTUDIO: FEBRUARY 2015

The TradersStudio code based on Dave Cline's article in this issue, "Candlesticks, Condensed," is provided at the following websites (and is shown below):

- www.TradersEdgeSystems.com/traderstips.htm
- www.TradersStudio.com → Traders Resources

The following code files are provided in the download:

- Function: CANDLE_SIG. Returns 1 if the candle signature could be computed and also by reference it sets the values for HO, HC & OL (the candle signature); returns zero if the signature could not be computed and sets each of HO, HC, & OL to -1.
- System: CANDLE_COND. A long-only system that uses weekly data and a single bar candle signature to enter at the next weekly open and exits as soon as the entered candle signature is other than the one input.

The system provided allows the user to explore the candle signatures using daily, weekly, or monthly data. I used weekly data on the full-sized S&P 500 futures contract (using data from Pinnacle Data Corp.) and optimized for the various signatures. One of the better ones was the 3:4:2 weekly candle signature (see Figure 8). I then added other index futures (DJ, DX, MD, ND, and RL) and ran the backtest with the 3:4:2 parameters, which resulted in the curves shown in Figure 9. The results shown in Figures 8 & 9 do not include slippage & commissions. Over the 15-year test period, there were only 50 trades for this index portfolio. Much more work is needed, but the candle signature provides a useful tool for researching candle patterns.

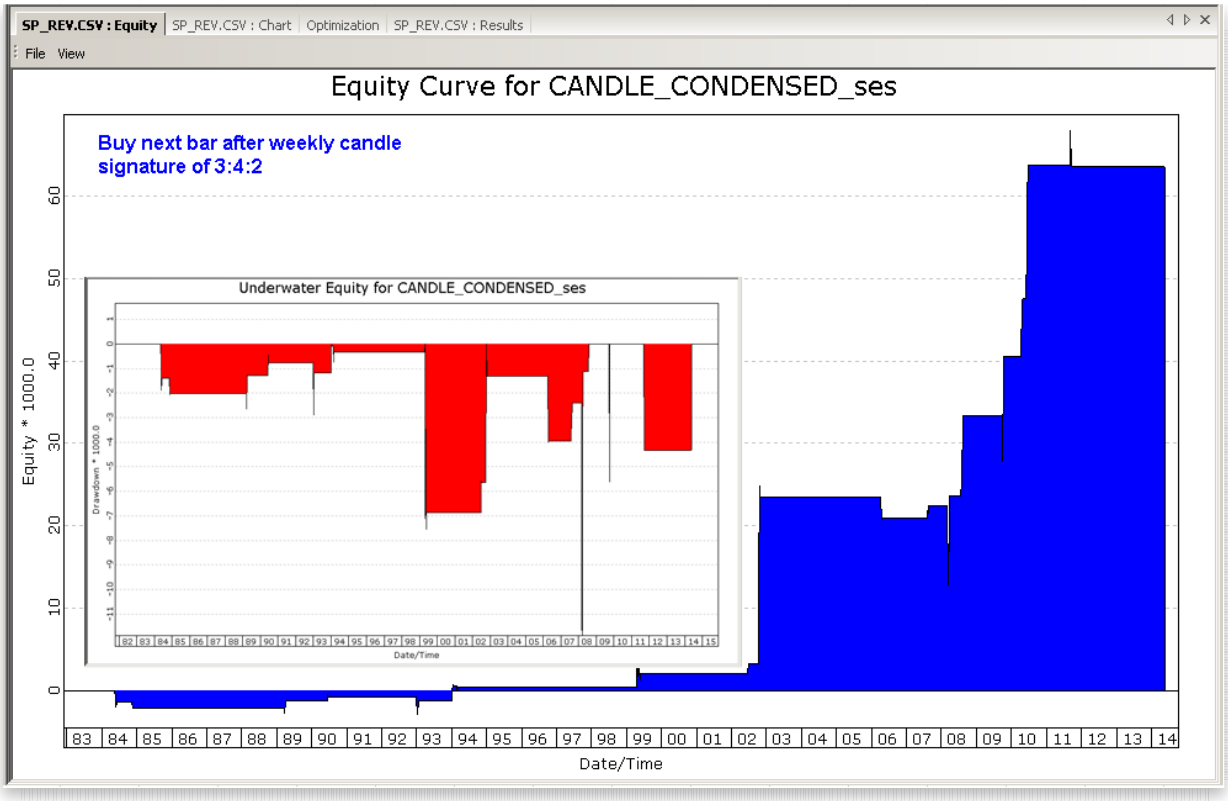


FIGURE 8: TRADERSSTUDIO, SAMPLE EQUITY CURVES FOR SYSTEM ON S&P CONTRACT. Here are sample equity and underwater equity curves for the CANDLE_CONDENSED system trading one contract per signal on the S&P 500 (SP) futures contract with parameters 3:4:2 from 1999 to 2014.

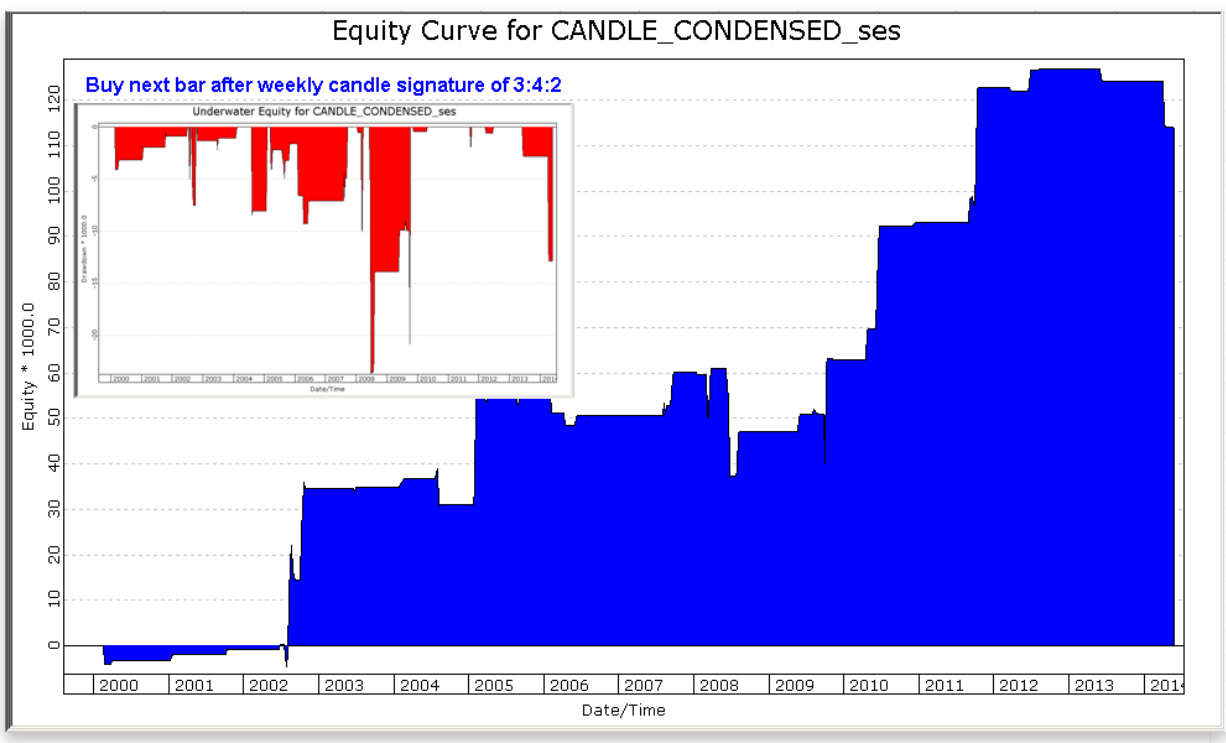


FIGURE 9: TRADERSSTUDIO, SAMPLE EQUITY CURVES FOR SYSTEM ON INDEX FUTURES PORTFOLIO OF SIX CONTRACTS. Here are sample equity and underwater equity curves for the CANDLE_CONDENSED system trading one contract per signal on the index futures portfolio of six contracts with parameters 3:4:2 from 1999 to 2014.

```
'CANDLESTICKS, CONDENSED
'Author: David Cline, TASC February 2015
'Coded by: Richard Denning, 12/10/2014
'www.TradersEdgeSystems.com

Function CANDLE_SIG(avgRngLen,SegCt,ByRef HO,ByRef HC,ByRef OL)
Dim avgRng As BarArray
avgRng = Average(H-L,avgRngLen)
Dim rngMult As BarArray
If avgRng <> 0 Then rngMult = (H-L)/avgRng
Dim segDiv
If SegCt <> 0 Then segDiv = 100/SegCt
Dim cndlRng
cndlRng = (H-L)/100
Dim rngMult2
rngMult2 = IIF(rngMult>1,1,rngMult)
If SegCt <> 0 And cndlRng <> 0 And segDiv <> 0 Then
    HO = Round((((H-O)/cndlRng)*rngMult2)/segDiv,0)
    HC = Round((((H-C)/cndlRng)*rngMult2)/segDiv,0)
    OL = Round((((O-L)/cndlRng)*rngMult2)/segDiv,0)
    CANDLE_SIG = 1
    Else
       HO=-1
       HC=-1
       OL=-1
       CANDLE_SIG = 0
End If
End Function
'-------------------------------------------------------------------
'SYSTEM TO TEST CANDLE SIGNATURE PATTERS:
Sub CANDLE_COND(hoS,hcS,olS,avgRngLen,SegCt)
Dim HO As BarArray
Dim HC As BarArray
Dim OL As BarArray
Dim CandleSigOK As BarArray
If BarNumber = FirstBar Then
    HO = -1
    HC = -1
    OL = -1
    CandleSigOK = 0
End If
CandleSigOK = CANDLE_SIG(avgRngLen,SegCt,HO,HC,OL)
If CandleSigOK=1 And HO=hoS And HC=hcS And OL=olS Then
    'Print FormatDateTime(Date)," ",GETACTIVESYMBOL(0)," ",HO,":",HC,":",OL
    Buy("LE",1,0,market,day)
End If
If CandleSigOK=1 And HO<>hoS And HC<>hcS And OL<>olS Then
    ExitLong("LX","",1,0,Market,Day)
```

```
End If
End Sub
'-----------------------------------------------------------------------------
```

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

UPDATA: FEBRUARY 2015

Our Traders' Tip for this month is based on "Candlesticks, Condensed" by Dave Cline in this issue.

In the article, Cline assigns a signature three-number classification system to traditional OHLC candlesticks, in order to better parameterize their OHLC distance ratios. He then uses the output as a predictor for future behavior.

The Updata code based on Cline's article can be found in the Updata library and may be downloaded by clicking the *custom* menu and *indicator library*. Those who cannot access the library due to a firewall may paste the code below into the Updata custom editor and save it.

```
'CandlesCondensed
PARAMETER "Avg. Period" #PERIOD=20
PARAMETER "Segment Count" #SEGMENTCOUNT=5
DISPLAYSTYLE LINE
INDICATORTYPE TOOL
NAME CANDLES CONDENSED
@SEGMENTDIVISOR=0
@RANGE=0
@RANGEMULT=0
@HO=0
@HC=0
@OL=0
@CANDLESIGNATURE=0
FOR #CURDATE=#PERIOD TO #LASTDATE
   @SEGMENTDIVISOR=100/#SEGMENTCOUNT
   @RANGEMULT=MIN(HIGH-LOW/SGNL(HIGH-LOW,#PERIOD,M),1)
   @RANGE=(HIGH-LOW)/100
   @HO=INT((@RANGEMULT/@SEGMENTDIVISOR)*((HIGH-OPEN)/@RANGE))
   @HC=INT((@RANGEMULT/@SEGMENTDIVISOR)*((HIGH-CLOSE)/@RANGE))
   @OL=INT((@RANGEMULT/@SEGMENTDIVISOR)*((OPEN-LOW)/@RANGE))
   DRAWTEXT LOW BELOW @HO ":" @HC ":" @OL
NEXT
```

A sample chart implementation is in Figure 10.

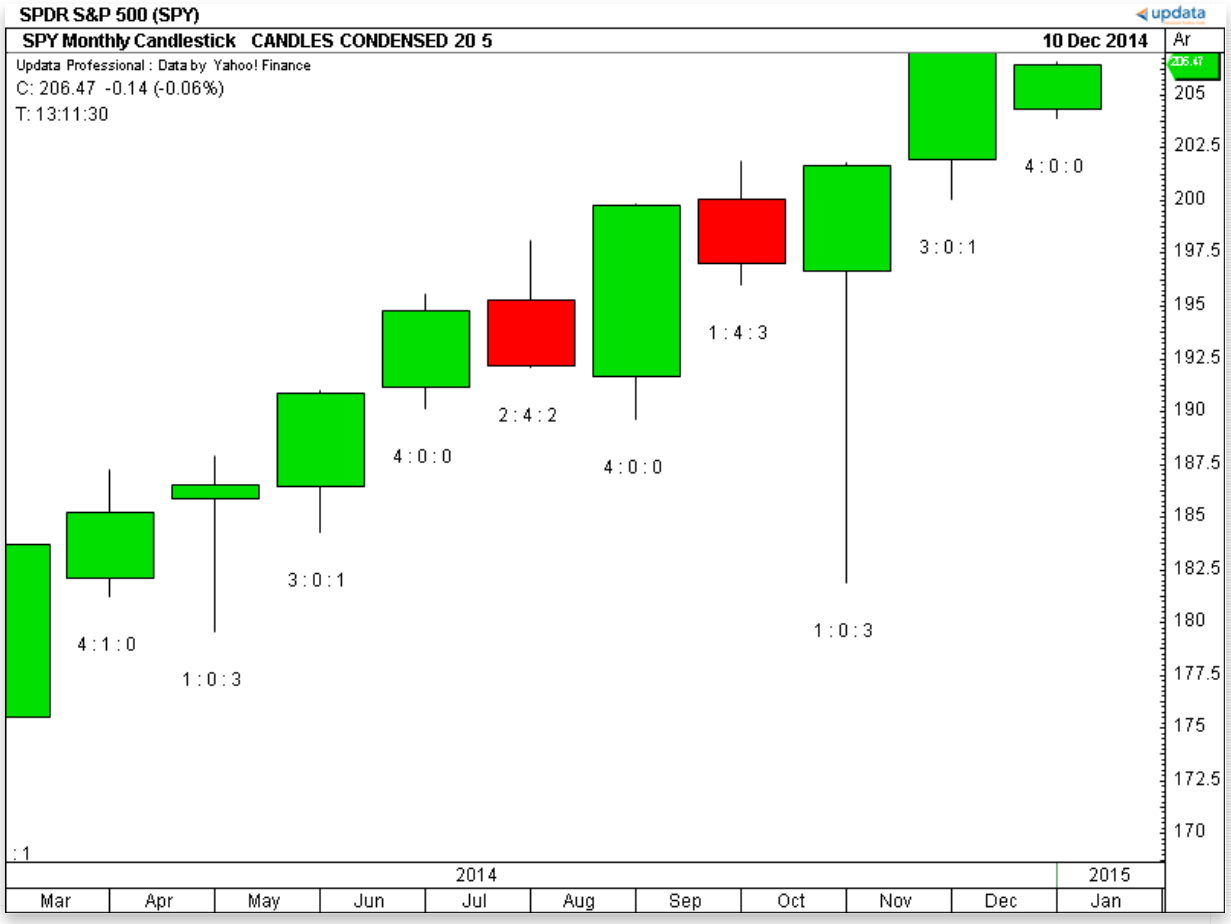


FIGURE 10: UPDATA. Here is the condensed candlestick technique as applied to the monthly SPY ETF. The number classification is displayed beneath the candlesticks, as opposed to a grid.

—Updata support team
support@updata.co.uk
www.updata.co.uk

BACK TO LIST

MICROSOFT EXCEL: FEBRUARY 2015

In “Candlesticks, Condensed” in this issue, author Dave Cline presents a way to construct candle signatures. He then suggests that you can use a database of these signatures to perform a statistical analysis of their predictive power as to market movements one, two, three, four, or more bars into the future.

Cline then suggests that the user might also combine a sequence of two or more consecutive candle signatures into a single key and perhaps gain a more deterministic probability output.

Cline's work in his article represents a step into the broader realm of time series data mining. A simple Google search on the words "time series meaningful pattern discovery" (or some variation thereof) will pull up quite a number of research papers that propose methods of data mining number or letter series to discover subsequences (words?) of various lengths that may have predictive power.

Cline's candle signature method creates a financial time series "alphabet" where the possible number of "letters" in this alphabet can be determined by user-defined resolution settings.

As Cline, and the literature, suggest, you will need a significant amount of storage and computing power to fully exploit the possibilities of finding "words" with statistical predictive value. Perhaps one day someone will figure out how to spell "approaching strong uptrend" with this alphabet! What an interesting way to start trying!

The Excel spreadsheet for this month uses the logic described in Cline's article and fleshed out in Cline's Python code to create the bar signatures. (Cline's Python code can be found at the S&C website from the *article code* link at www.traders.com.) Cline's code accumulates an output database of unique candle signatures found by cycling through bars for 10 different sector ETFs.

The spreadsheet I am providing here will allow you to use Cline's ETF list or any other set of symbols with historical data on Yahoo! Finance. You may use as many symbols as you have the patience to process.

Figure 11 shows the controls for building the signature database. You set the items that are bolded in blue to your tastes and click the button.

Controls for Condensed Candle data base building process.

Build
Condensed Candle Signatures
From Sourcing Securities

Data Type	**Daily**	
	09:58:13.09	Start time
	09:59:13.09	End time
	00:01:00.00	**Elapsed**
Data base build user controls		
10	Weight Periods (SMA Length)	
4	Segment Count	
	Total Bars Evaluated:	38,993
10	**Signature Sourcing Securities**	
Symbol	**Name (Symbol) -Market (Frequency)**	**Bar Count**
XLB	Materials Select (XLB) -PCX (Daily)	4,019
XLE	SPDR Select Secto (XLE) -PCX (Daily)	4,019
XLF	SPDR Select Secto (XLF) -PCX (Daily)	4,019
XLI	SPDR Select Secto (XLI) -PCX (Daily)	4,019
XLK	SPDR Select Secto (XLK) -PCX (Daily)	4,019
XLP	SPDR Select Secto (XLP) -PCX (Daily)	4,019
XLU	SPDR Select Secto (XLU) -PCX (Daily)	4,019
XLV	SPDR Select Secto (XLV) -PCX (Daily)	4,019
XLY	SPDR Select Secto (XLY) -PCX (Daily)	4,019
AGG	iShares Core U.S. (AGG) -PCX (Daily)	2,822

You may specify as many symbols as you like in column A.
But the more you specify, the longer the build will take.
Do not leave gaps. The first empty or blank cell in the symbol column stops the scan process

The elapsed time includes the time it takes to download the historical data for each symbol and the necessary calculation time.

The weight periods value has little effect on run time but can change the way a bar is characterized

Segment count can have a dramatic effect on the number of bars found and some impact on the elapsed time required to find them.

FIGURE 11: EXCEL, BUILDING THE SIGNATURE

DATABASE

After plenty of screen flashing for a minute or so, we will have built our signature statistics "database" on the CondensedCandleStats tab (Figure 12). We also have our signature database sorted by rank at the various look-forward periods on the RankedCandleStats tab (Figure 13).

Daily Candlestick Signature Index with Statistics

77 Unique Signatures encountered out of the 125 signatures that are possible when we use a segment count setting of 4.

10 Symbols Evaluated
38,993 Bars Evaluated
00:01:00.00 Elapsed

Look Forward

1 Bar Look Forward Results						2 Bar Look Forward Results						3 Bar Look Forward Results						4 Bar Look Forward Results					
Signature	Bar Count	Rank	UPs as % of Bars	Avg $ Return	Avg % Return	Signature	Bar Count	Rank	% of Bars UP	Avg $ Return	Avg % Return	Signature	Bar Count	Rank	% of Bars UP	Avg $ Return	Avg % Return	Signature	Bar Count	Rank	% of Bars UP	Avg $ Return	Avg % Return
0_0_0	4	-7.43	25.00	-0.17	-0.30	0_0_0	4	-20.71	50.00	-0.15	-0.21	0_0_0	4	-11.94	25.00	-0.19	-0.48	0_0_0	4	27.03	50.00	0.00	0.27
0_0_1	56	7.46	44.64	-0.02	0.01	0_0_1	56	184.59	57.14	0.01	0.10	0_0_1	56	274.84	53.57	0.03	0.17	0_0_1	56	569.65	55.36	0.08	0.33
0_0_2	250	-1,133.85	48.00	-0.05	-0.20	0_0_2	250	-487.39	55.20	-0.02	-0.06	0_0_2	250	-1,713.92	54.40	-0.04	-0.23	0_0_2	250	-2,054.87	54.00	-0.06	-0.28
0_0_3	201	-303.64	45.27	-0.03	-0.07	0_0_3	201	74.21	46.77	0.00	0.02	0_0_3	201	689.62	48.76	0.04	0.14	0_0_3	201	1,011.11	52.24	0.05	0.18
0_0_4	120	-369.56	48.33	-0.04	-0.13	0_0_4	120	-461.23	50.83	-0.03	-0.15	0_0_4	120	-1,514.18	48.33	-0.13	-0.54	0_0_4	120	-1,730.14	49.17	-0.12	-0.60
0_1_1	108	268.53	56.48	0.02	0.08	0_1_1	108	232.79	50.00	0.00	0.09	0_1_1	108	-40.94	49.07	-0.07	-0.02	0_1_1	108	315.43	50.93	-0.03	0.11
0_1_2	674	-969.98	46.59	-0.02	-0.07	0_1_2	674	-1,334.76	50.00	-0.02	-0.08	0_1_2	674	-1,664.29	48.66	-0.03	-0.10	0_1_2	674	-1,554.26	50.59	-0.03	-0.09
0_1_3	660	128.91	51.06	0.00	0.01	0_1_3	660	-1,430.01	50.61	-0.03	-0.08	0_1_3	660	617.69	52.73	0.01	0.03	0_1_3	660	699.62	52.12	0.01	0.04

FIGURE 12: EXCEL, SIGNATURE DATABASE WITH PERTINENT STATISTICS FOR FOUR LOOK-FORWARD BAR COUNTS

Daily Candlestick Signatures by Rank

77 Unique Signatures encountered out of the 125 signatures that are possible when we use a segment count setting of 4.

10 Symbols Evaluated
38,993 Bars Evaluated
00:01:00.00 Elapsed

Look Forward

1 Bar Look Forward Results						2 Bar Look Forward Results						3 Bar Look Forward Results						4 Bar Look Forward Results					
Signature	Bar Count	Rank	UPs as % of Bars	Avg $ Return	Avg % Return	Signature	Bar Count	Rank	% of Bars UP	Avg $ Return	Avg % Return	Signature	Bar Count	Rank	% of Bars UP	Avg $ Return	Avg % Return	Signature	Bar Count	Rank	% of Bars UP	Avg $ Return	Avg % Return
1_4_3	1,186	9,352.05	58.52	0.08	0.23	0_4_4	1,666	12,195.67	56.36	0.06	0.23	0_4_4	1,666	15,762.90	58.34	0.07	0.28	1_3_3	1,544	15,091.27	58.61	0.08	0.28
0_4_4	1,671	9,208.15	55.72	0.05	0.18	1_3_3	1,545	9,517.99	56.83	0.06	0.19	1_4_3	1,183	11,309.51	55.79	0.12	0.31	2_3_2	794	12,888.02	57.68	0.15	0.49
2_3_2	795	4,863.61	55.47	0.06	0.20	2_3_2	795	9,317.40	58.87	0.11	0.34	2_4_2	523	10,896.73	60.23	0.14	0.57	1_2_2	1,054	11,565.54	57.97	0.09	0.33
1_3_3	1,545	4,212.64	55.79	0.03	0.09	1_4_3	1,183	8,195.14	55.37	0.09	0.23	2_3_2	795	10,840.94	59.62	0.13	0.38	1_4_3	1,183	11,248.56	57.06	0.11	0.29
2_4_2	523	4,208.21	59.66	0.07	0.23	2_4_2	523	5,805.30	56.21	0.08	0.35	1_3_3	1,545	9,929.14	55.99	0.06	0.21	0_4_4	1,664	10,728.94	55.95	0.05	0.21
2_1_2	979	3,771.63	53.73	0.05	0.13	2_2_1	867	4,585.62	55.36	0.06	0.17	1_2_2	1,054	6,795.37	55.03	0.05	0.21	2_4_2	523	10,257.90	56.79	0.15	0.61
2_2_1	867	3,705.70	55.25	0.05	0.14	1_2_3	942	4,581.53	54.25	0.02	0.17	0_3_3	597	5,363.60	55.11	0.08	0.30	0_3_3	597	6,638.80	53.77	0.11	0.38
3_2_1	680	3,675.57	55.00	0.06	0.18	3_3_0	164	3,896.10	60.37	0.15	0.65	4_1_0	1,477	5,096.46	53.08	0.03	0.12	2_2_1	867	6,225.76	55.48	0.07	0.23
1_3_2	555	2,331.08	53.69	0.05	0.15	1_2_2	1,054	3,853.43	54.46	0.03	0.12	1_2_3	941	4,968.01	55.47	0.03	0.17	2_3_1	353	5,924.45	58.36	0.13	0.49
0_3_3	597	2,194.60	52.93	0.02	0.13	2_3_1	353	3,731.84	57.51	0.07	0.32	2_2_1	867	4,619.10	55.94	0.05	0.17	1_2_3	941	5,714.01	53.99	0.03	0.21

FIGURE 13: EXCEL, RANKED SIGNATURES BY LOOK-FORWARD BAR COUNT

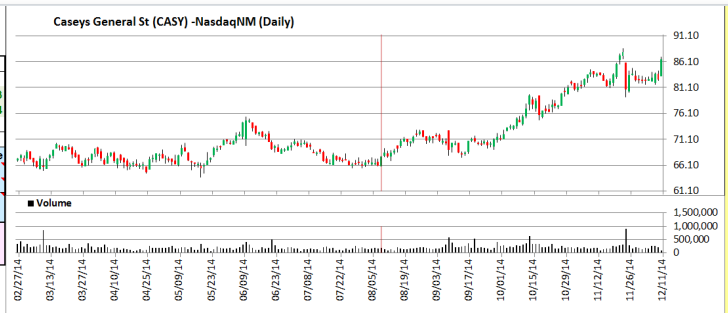
To put the signature database to use, the *CalculationsAndCharts* tab (Figure 14) calculates the bar signatures for the symbol shown on the chart. The signature for the bar at the cursor is used to look up and display values from the signature data at the bottom of the chart.

Candlesticks Condensed
Caseys General St (CASY) -NasdaqNM (Daily)

User Charting Controls
Data Control
7855 Available Input Data Rows — Earliest: **10/20/1983**
Use Close Adjusted: **TRUE** — Latest: **12/11/2014**
2800 Logically Available for Calc / Plot

Price Chart Windowing Controls
0 Input Data Offset (Pan)
201 Data Points to Plot (Zoom)
Manual <= Set Chart
First: **Thu 02/27/2014**
Last: 10:37am Thu 12/11/2014
Y-axis Range — **Max High+ 91.30** — **Min Low- 61.10**
White space %: **5%**
Allowance: 1.30

Help! I would like to plot data
From **10/04/2013** **201**
To **07/23/2014** **99**
<== Chart These Dates



To use:
1. **Go to CandleStatControls tab to begin building the Signatures Data base. Select stocks and index symbols you would like to use to populate the signatures DB.**
2. **After building the signatures data base (see CondensedCandleStats tab), go to the Follow or the InputPriceData tab to select a symbol you would like to evaluate with respect to the signatures you have accumulated in the DB and to asses their predictive capabilities with respect to your stock.**

Date and prices @ Cursor => 88(RR) 08/08/2014 186,000(v) 65.97(o) 67.87(h) 65.88(l) 67.73(c)

Candle Signature for bar at cursor:	4_0_0	DB Statistics for this Signature ==>	@ Bars Forward	DB Bars That Match Pattern	% of time Forward Bar was up	Average % Gain
Candle DB Index for this Signature:	60		1	175	48.00	-0.0697
			2	175	51.43	-0.0423
			3	175	50.86	-0.1637
			4	175	51.43	-0.1144

Candle Signature Calculations for Caseys General St (CASY) -NasdaqNM (Daily)

Rel Row	Stock Prices Date	Volume	Open	High	Low	Close	P Cursor 91.30	P Cursor Min 61.10	Vol Cursor 896,200	P Cursor Min 0	Candle Range	SMA(10) of Candle Range	Range Multiplier	Capped Range Multiplier	Factor	HO Range	HO	HC Range	H
1	Thu 12/11/2014	103,711	83.38	87.00	83.38	86.55	88	91.30	88	986,200	3.62	1.8200	1.99	1.00	1.1050	3.62	4	0.45	
2	Wed 12/10/2014	226,700	83.68	84.32	82.43	82.70	88	61.10	88	0	1.89	1.8760	1.01	1.00	2.1164	0.64	1	1.62	

FIGURE 14: EXCEL, SIGNATURE. What does the signature database have to say about the bar under the cursor?

The spreadsheet file for this Traders' Tip can be downloaded below. To successfully download it, follow these steps:

- Right-click on the Excel file link, then
- Select "save as" (or "save target as") to place a copy of the spreadsheet file on your hard drive.

Note: Dave Cline's Python code provided with his article appears to have been written to run on the Quantopian website platform. I am providing a link to that site for reference only: https://www.quantopian.com/home.

—Ron McAllister
Excel and VBA programmer
rpmac_xltt@sprynet.com

BACK TO LIST

Originally published in the February 2015 issue of
Technical Analysis of STOCKS & COMMODITIES magazine.

March 2015

For this month's Traders' Tips, the focus is John Ehlers & Ric Way's article in this issue, "Trading System Design: A Statistical Approach." Here, we present the March 2015 Traders' Tips code with possible implementations in various software.

Code in EasyLanguage is already provided by Ehlers & Way in their article, which S&C subscribers will find in the Subscriber Area of our website here.

The Traders' Tips section is provided to help the reader implement a selected technique from an article in this issue or another recent issue. The entries here are contributed by software developers or programmers for software that is capable of customization.

TRADESTATION: MARCH 2015

In "Trading System Design: A Statistical Approach" in this issue, authors John Ehlers & Ric Way outline a procedure for the development of trading systems using a statistical approach. In the article, they create a set of data that they analyze using Microsoft Excel spreadsheet software. They have provided TradeStation EasyLanguage code for an indicator to help create the data for analysis, as well as a simple test strategy to demonstrate the process.

To download the EasyLanguage code, please visit our TradeStation and EasyLanguage support forum. The code for this article can be found here: http://www.tradestation.com/TASC-2015, and is also shown below. The ELD filename is "_TASC_StatisticalApproach.ELD."

For more information about EasyLanguage in general, please see http://www.tradestation.com/EL-FAQ.

A sample chart is shown in Figure 1.

FIGURE 1: TRADESTATION. Here is an example of a simple stochastic system applied to a daily chart of the emini S&P 500 (ES), based on John Ehlers & Ric Way's article in this issue.

This article is for informational purposes. No type of trading or investment recommendation, advice, or strategy is being made, given, or in any manner provided by TradeStation Securities or its affiliates.

Indicator: _Ehlers Event Predictability

```
{
_Ehlers Event Predictability Indicator

For application see..
Technical Analysis of Stocks and Commodities
March 2015
```

```
}

variables:
        Event( false ),
        FuturePrice( 0 ),
        j( 0 ),
        CG( 0 ),
        Denom( 0 ) ;
arrays:
        PredictBin[100]( 0 );


//>>>>>>>>>> Start Event Code
inputs:
        StocLength( 10 ) ;
variables:
        HiC( 0 ),
        LoC( 0 ),
        Stoc( 0 ) ;
HiC = Highest( Close, StocLength ) ;
LoC = Lowest( Close, StocLength ) ;
Stoc = ( Close - LoC ) / ( HiC - LoC );
if Stoc[9] crosses under 0.2 then
        Event = true
else
        Event = false ;
//<<<<<<<<<<<< End Event Code


If Event then
        begin
        FuturePrice = 100 * ( Close - Close[9] ) / Close[9] ;
        //Future is referenced to 10 bars back

        If FuturePrice < -10 then
                FuturePrice = -10 ; //Limits lower price to -10%

        If FuturePrice > 10 then
                FuturePrice = 10 ; //Limits higher price to +10%

        FuturePrice = 5 * ( FuturePrice + 10 ) ;
        //scale -10% to +10% to be 0 - 100
        end ;

//Place the FuturePrices into one of 100 bins
If FuturePrice <> FuturePrice[1] then
        begin
        For j = 1 to 100
                begin
                If FuturePrice > j - 1 and FuturePrice <= j then
                        PredictBin[j] = PredictBin[j] + 1 ;
                end;
        end;

//Measure Center of Gravity as a quick estimate
CG = 0 ;
Denom = 0 ;
For j = 1 to 100
        begin
        CG = CG + j * PredictBin[j] ;
```

```
        Denom = Denom + PredictBin[j] ;
        end ;

CG = (CG/Denom-50)/5;

Plot1( CG, "CG" );

if LastBarOnChartEx then
        begin
        For j = 0 to 100
                begin
                Print( File( "C:\PDFTest\PDF.CSV" ),
                .2*j - 10, ",", PredictBin[j] ) ;
                end ;
        end ;


Strategy: _Ehlers Stochastic Strategy

{
_Ehlers Stochastic Strategy
Technical Analysis of Stocks and Commodities
March 2015
}

inputs:
        StocLength( 8 ),
        Threshold( .3 ),
        TradeLength( 14 ),
        PctLoss( 3.8 ) ;

variables:
        HiC( 0 ),
        LoC( 0 ),
        Stoc( 0 ) ;

HiC = Highest( Close, StocLength ) ;
LoC = Lowest( Close, StocLength ) ;

Stoc = ( Close - LoC ) / ( HiC - LoC ) ;

If Stoc crosses under Threshold then
        Buy next bar on Open ;
If Barssinceentry >= TradeLength then
        Sell next bar on Open ;
If Low < EntryPrice * ( 1 - PctLoss / 100 ) then
        Sell next bar on Open ;
```

—Doug McCrary
TradeStation Securities, Inc.
www.TradeStation.com

BACK TO LIST

eSIGNAL: MARCH 2015

For this month's Traders' Tip, we're providing the formula SimpleStocTrSystem.efs based on the formula described in John Ehlers & Ric Way's article in this issue, "Trading System Design: A Statistical Approach."

The study contains formula parameters that may be configured through the *edit chart* window (right-click on the chart and select "edit chart"). A sample chart is shown in Figure 2.

FIGURE 2: eSIGNAL. Here is an example of the simple stochastic system on a chart of the S&P 500 emini futures contract (ES).

To discuss this study or download a complete copy of the formula code, please visit the EFS Library Discussion Board forum under the forums link from the support menu at www.esignal.com or visit our EFS KnowledgeBase at http://www.esignal.com/support/kb/efs/. The eSignal formula script (EFS) is also available below.

SimpleStocTrSystem.efs

```
/**********************************
Provided By:
    Interactive Data Corporation (Copyright © 2015)
    All rights reserved. This sample eSignal Formula Script (EFS)
    is for educational purposes only. Interactive Data Corporation
    reserves the right to modify and overwrite this EFS file with
    each new release.

Description:
    Trading System Design: A Statistical Approach by John F. Ehlers and Ric Way

Version:            1.00  01/12/2015

Formula Parameters:                     Default:
Stoc Length                             8
Threshold                               0.3
Trade Length                            14
Percent Loss                            3.8
Entry Position Color                    lime
Exit Position Color                     red

Notes:
    The related article is copyrighted material. If you are not a subscriber
    of Stocks & Commodities, please visit www.traders.com.

**********************************/

var fpArray = new Array();

function preMain(){
```

```
setStudyTitle("SimpleStocTrSystem");
setPriceStudy(true);

var x = 0;

fpArray[x] = new FunctionParameter("fpLength", FunctionParameter.NUMBER);
with(fpArray[x++]){
    setName("Stoc Length");
    setLowerLimit(1);
    setDefault(8);
}

fpArray[x] = new FunctionParameter("fpThreshold", FunctionParameter.NUMBER);
with(fpArray[x++]){
    setName("Threshold");
    setLowerLimit(0);
    setUpperLimit(1);
    setDefault(0.3);
}

fpArray[x] = new FunctionParameter("fpTradeLength", FunctionParameter.NUMBER);
with(fpArray[x++]){
    setName("Trade Length");
    setLowerLimit(0);
    setDefault(14);
}

fpArray[x] = new FunctionParameter("fpPctLoss", FunctionParameter.NUMBER);
with(fpArray[x++]){
```

```
        setName("Percent Loss");
        setLowerLimit(0);
        setUpperLimit(100);
        setDefault(3.8);
    }

    fpArray[x] = new FunctionParameter("fpEntryColor", FunctionParameter.COLOR);
    with(fpArray[x++]){
        setName("Entry Position Color");
        setDefault(Color.lime);
    }

    fpArray[x] = new FunctionParameter("fpExitColor", FunctionParameter.COLOR);
    with(fpArray[x++]){
        setName("Exit Position Color");
        setDefault(Color.red);
    }
}

var bInit = false;
var bVersion = null;

var xClose = null;
var xOpen = null;
var xLow = null;

var xStoc = null;

var nBarsSinceEntry = null;
var nEntryPrice = null;
```

```
var nLotSize = null;


function main(fpLength, fpThreshold, fpTradeLength, fpPctLoss, fpEntryColor,
fpExitColor){


    if (bVersion == null) bVersion = verify();

    if (bVersion == false) return;


    if (!bInit){


        xClose = close();

        xOpen = open();

        xLow = low();


        xStoc = efsInternal("calc_Stoc", fpLength, xClose);


        nLotSize = Strategy.getDefaultLotSize();


        bInit = true;

    }


    if (getCurrentBarIndex() != 0){


        if (!Strategy.isInTrade() && crossUnder(xStoc, fpThreshold, fpLength)){


            nEntryPrice = xOpen.getValue(1);


            Strategy.doLong("Enter Long", Strategy.MARKET,
Strategy.NEXTBAR);


            drawShapeRelative(1, BelowBar1, Shape.UPTRIANGLE, null, fpEntryColor,
```

```
Text.PRESET, getCurrentBarIndex()+"Ent");

            drawTextRelative(1, BelowBar2, "Enter Long", fpEntryColor, null,
Text.PRESET|Text.CENTER, null, null, getCurrentBarIndex()+"Ent_Label");

            drawTextRelative(1, BelowBar3, nLotSize + " @ " +
formatPriceNumber(nEntryPrice), fpEntryColor, null, Text.PRESET|Text.CENTER, null,
null, getCurrentBarIndex()+"Ent_Size");



            nBarsSinceEntry = getCurrentBarCount() + 1;



            return;

        }



        if (Strategy.isLong()){



            var bExit = false;

            var sLabel = "";



            var nCountEntry = getCurrentBarCount() - nBarsSinceEntry;

            if (nCountEntry >= fpTradeLength){

                bExit = true;

                sLabel = "Exit Length";

            }



            var nLow = xLow.getValue(0);

            var nStopLevel = nEntryPrice * (1 - fpPctLoss / 100);

            if (nLow < nStopLevel){

                bExit = true;

                sLabel = "Exit Stop";

            }



            if (bExit){

                var nExitPrice = xOpen.getValue(1);
```

```
                Strategy.doSell("Exit Long", Strategy.MARKET, Strategy.NEXTBAR)

                drawShapeRelative(1, AboveBar1, Shape.DOWNTRIANGLE, null,
fpExitColor, Text.PRESET, getCurrentBarIndex()+"Ex");

                drawTextRelative(1, AboveBar2, sLabel, fpExitColor, null,
Text.PRESET|Text.CENTER, null, null, getCurrentBarIndex()+"Ex_Label");

                drawTextRelative(1, AboveBar3, nLotSize + " @ " +
formatPriceNumber(nExitPrice), fpExitColor, null, Text.PRESET|Text.CENTER, null,
null, getCurrentBarIndex()+"Ex_Size");

            }

        }

    }

}



var xHignest = null;

var xLowest = null;



function calc_Stoc(nLength, xSource){



   if (getBarState() == BARSTATE_ALLBARS){

       xHignest = highest(nLength, xSource);

       xLowest = lowest(nLength, xSource);

   }



   var nSource = xSource.getValue(0);

   var nHighest = xHignest.getValue(0);

   var nLowest = xLowest.getValue(0);



   if (nSource == null || nHighest == null || nLowest == null)

        return;



   var nReturnValue = (nSource - nLowest) / (nHighest - nLowest);
```

```
   return nReturnValue;
}

function crossUnder(xStoc, nThreshold, nStocLength){

    var nReturnValue = false;

    var nStoc = xStoc.getValue(0);

    if (nStoc < nThreshold){
        for (var i = -1; i >= -(getCurrentBarCount() - nStocLength) ; i--){

            var nPrevStoc = xStoc.getValue(i);

            if (nPrevStoc != nThreshold){
                if (nPrevStoc > nThreshold)
                    nReturnValue = true;
                break;
            }
        }
    }
    return nReturnValue;
}

function verify(){

    var b = false;

    if (getBuildNumber() < 779){
```

```
        drawTextAbsolute(5, 35, "This study requires version 8.0 or later.",

            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,

            null, 13, "error");

        drawTextAbsolute(5, 20, "Click HERE to upgrade.@URL=http://www.esignal.com
/download/default.asp",

            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,

            null, 13, "upgrade");

        return b;

    }

    else{

        b = true;

    }



    return b;

}
```

—Eric Lippert
eSignal, an Interactive Data company
800 779-6555, www.eSignal.com

BACK TO LIST

WEALTH-LAB: MARCH 2015

In their article in this issue, "Trading System Design: A Statistical Approach," authors John Ehlers & Ric Way outline a statistically valid procedure for the successful development of trading systems, providing a testbed for assessing whether the price will increase or decrease over *n* bars after an event.

From our point of view, it might be optimal to prove the conclusion regarding the robustness of the example system by using a different subset of data that includes a bear market, given that the in-sample period of 10 years used to optimize the system on was a strong bull market (Figure 3).

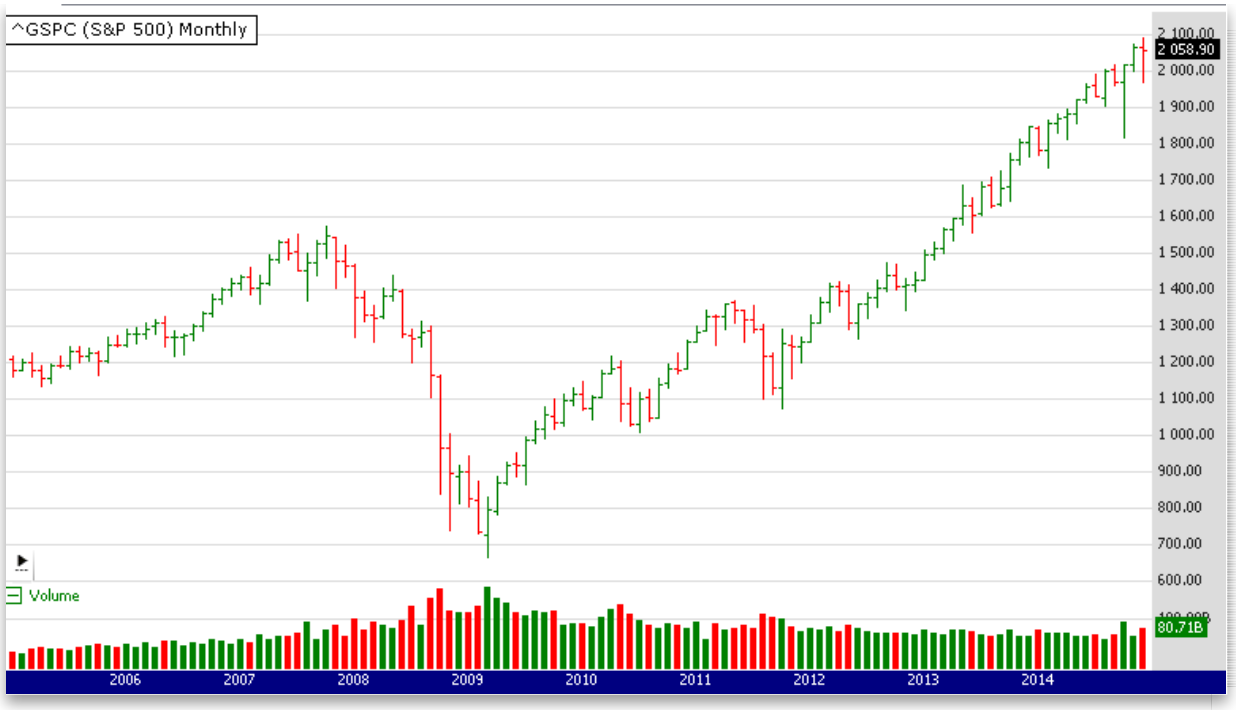


FIGURE 3: WEALTH-LAB. This chart shows the US market bubble of the 2010s on a monthly chart of the S&P 500 index (∧GSPC).

The code for Wealth-Lab based on Ehlers & Way's code follows:

Wealth-Lab 6 strategy code (C#)

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Drawing;
using WealthLab;
using WealthLab.Indicators;

namespace WealthLab.Strategies
{
        public class EhlersMar2015 : WealthScript
        {
                private StrategyParameter paramStoc;
                private StrategyParameter paramThresh;
                private StrategyParameter paramLength;
                private StrategyParameter paramLoss;

                public EhlersMar2015()
                {
                        paramStoc = CreateParameter("StocLength", 8, 1, 100, 1);
                        paramThresh = CreateParameter("Threshold", 0.3, 0.1, 0.9,
0.1);
                        paramLength = CreateParameter("TradeLength", 14, 1, 50, 1);
                        paramLoss = CreateParameter("PctLoss", 3.8, 0.5, 15.0, 0.5);
                }

                protected override void Execute()
```

```
                {
                        int StocLength = paramStoc.ValueInt, TradeLength =
paramStoc.ValueInt;
                        double Threshold = paramThresh.Value, PctLoss =
paramLoss.Value;

                        Highest HiC = Highest.Series(Close, StocLength);
                        Lowest LoC = Lowest.Series(Close, StocLength);
                        DataSeries Stoc = (Close - LoC) / (HiC - LoC);

                        for(int bar = GetTradingLoopStartBar(1); bar < Bars.Count;
bar++)
                        {
                                if (IsLastPositionActive)
                                {
                                        Position p = LastPosition;
                                        if ( bar+1 - p.EntryBar >= TradeLength )
                                                SellAtMarket( bar+1, p, "Timed" );
                                        else
                                                if( Low[bar] < p.EntryPrice*(1.0 -
PctLoss /100d) )
                                                SellAtMarket(bar+1, p, "Stop");

                                }
                                else
                                {
                                        if( CrossOver( bar, Stoc, Threshold ) )
                                                BuyAtMarket( bar+1 );
                                }
                        }
                }
        }
}
```

—Eugene, Wealth-Lab team
MS123, LLC
www.wealth-lab.com

BACK TO LIST

NEUROSHELL TRADER: MARCH 2015

The simple stochastic trading system described by John Ehlers & Ric Way in their article in this issue, “Trading System Design: A Statistical Approach,” can be easily implemented with a few of NeuroShell Trader’s 800+ indicators. Simply select “New Trading Strategy” from the *insert* menu and enter the following in the appropriate locations of the Trading Strategy Wizard:

```
BUY LONG CONDITIONS:    CrossBelow(Stoch%K(High,Low,Close,5),30)
LONG TRAILING STOP:     PriceFloor%(Trading Strategy,3.8)
```

```
SELL LONG CONDITIONS:   BarsSinceFill>=X(Trading Strategy,14)
```

If you have NeuroShell Trader Professional, you can also choose whether the parameters should be optimized. After backtesting the trading strategy, use the *detailed analysis* button to view the backtest and trade-by-trade statistics for the strategy.

You can also create another trading strategy using the center of gravity indicator referenced in the article along with a one-period lag of the same indicator called the *trigger*. Both indicators are part of **Ehlers' Cybernetic Analysis** add-on for NeuroShell Trader.

```
BUY LONG CONDITIONS:    Center of Gravity > Center of Gravity Trigger
SELL LONG CONDITIONS:   Center of Gravity < Center of Gravity Trigger
```

Users of NeuroShell Trader can go to the STOCKS & COMMODITIES section of the NeuroShell Trader free technical support website to download a copy of this or any previous Traders' Tips.

A sample chart is shown in Figure 4.

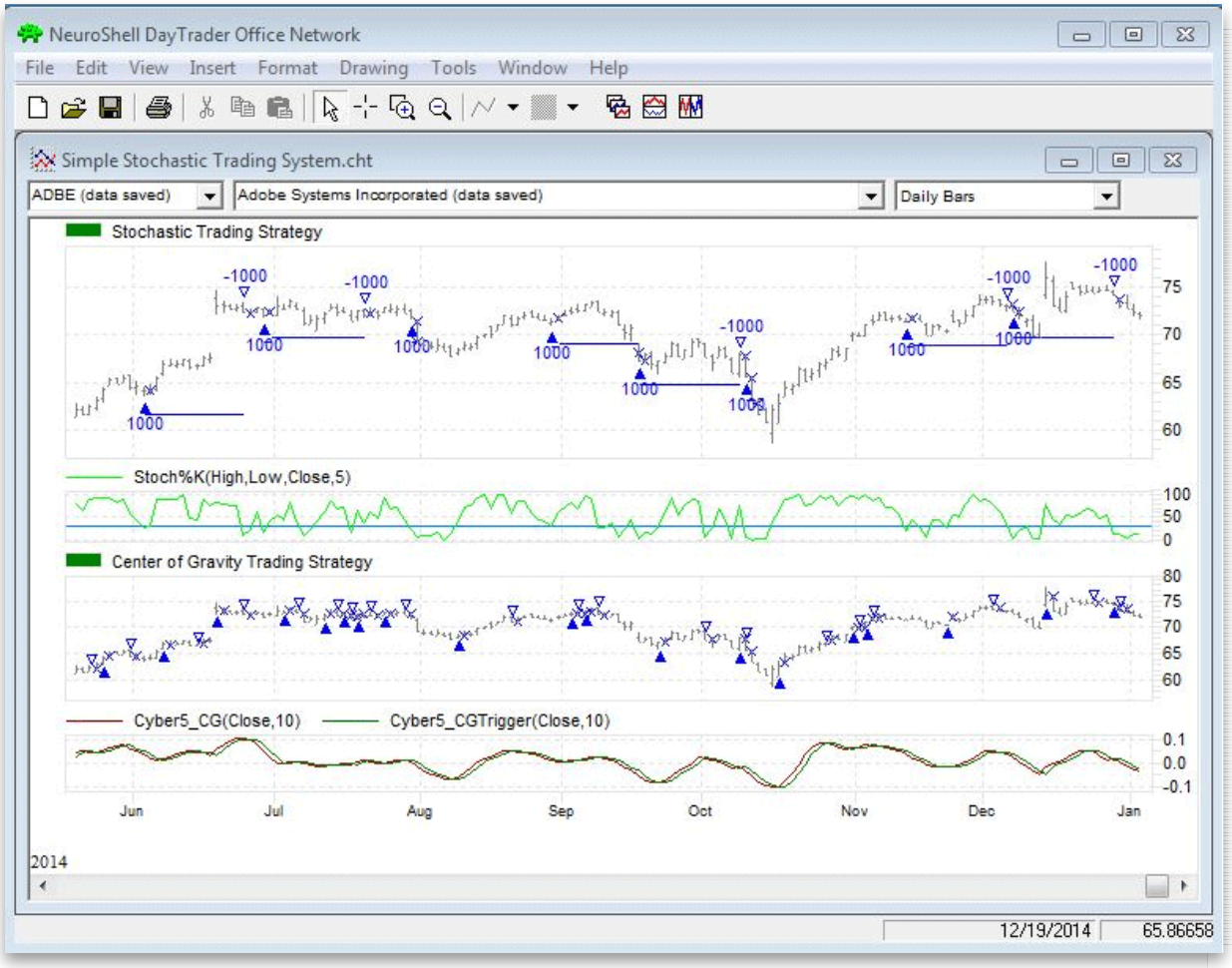


FIGURE 4: NEUROSHELL TRADER. This NeuroShell Trader chart displays the simple stochastic trading system as well as a trading system based on Ehlers' center of gravity indicator.

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

BACK TO LIST

AIQ: MARCH 2015

The AIQ code based on John Ehlers & Ric Way's article in this issue, "Trading System Design: A Statistical Approach," is provided at www.TradersEdgeSystems.com/traderstips.htm and is also shown here:

```
!TRADING SYSTEM DESIGN: A STATISTICAL APPROACH

!Author: John Ehlers, TASC March 2015

!Coded by: Richard Denning 1/12/2015

!www.TradersEdgeSystems.com


!STOCHASTIC TRADING SYSTEM FROM ARTICLE:


!INPUTS:

StocLength is 8.

Threshold is 0.3.

TradeLength is 14.

PctLoss is 3.8.


!SYSTEM CODE:

HiC is Highresult([Close], StocLength, 0).

LoC is Lowresult([Close], StocLength, 0).

Stoc is ([Close] - LoC) / (HiC - LoC).

Buy if Stoc < Threshold and valrule(Stoc >= Threshold,1).
```

```
PD is {position days}.

PEP is {position entry price}.

ExitLong if PD - 1 >= TradeLength or [Low] < PEP*(1-PctLoss/100).
```

Figure 5 shows the EDS backtest summary for trading the NASDAQ 100 list of stocks using the authors' stochastic system over the period 2009 through 1/13/2015.

Ehlers Systems Buy

		Winners	Losers
		==========	==========
Number of trades in test:	6781	3100	3663
Average periods per trade:	15.08	21.16	9.95
Maximum Profit/Loss:		85.28 %	(33.25)%
Average Drawdown:	(2.57)%	(0.89)%	(4.01)%
Average Profit/Loss:	1.06 %	6.75 %	(3.76)%
Average SPX Profit/Loss:	0.60 %	2.49 %	(1.00)%
Probability:		45.72 %	54.02 %
Average Annual ROI:	25.59 %	116.53 %	(137.85)%
Annual SPX (Buy & Hold):	20.33 %		
Reward/Risk Ratio:	1.52		
Start test date:	01/09/09		
End test date:	01/13/15		

Interval: Daily
Using list: NAS100
Includes Open Positions (1)
Pricing Summary
  Entry price: [Open]
  Exit price: [Open]
Exit Summary
  On Rule ExitLong (1)

FIGURE 5: AIQ. Here is the strategy's EDS backtest summary for trading the NASDAQ 100 list of stocks over the period from 2009 through 1/13/2015.

—Richard Denning
info@TradersEdgeSystems.com
for AIQ Systems

BACK TO

LIST

TRADERSSTUDIO: MARCH 2015

The TradersStudio code for John Ehlers & Ric Way's article in this issue, "Trading System Design: A Statistical Approach" can be found at:

- www.TradersEdgeSystems.com/traderstips.htm
- www.TradersStudio.com → Traders Resources

The following code file is provided in the download:

- System: EHLERS_SYSTEMS: A long-only system that uses daily data and the stochastic indicator for entries.

Figure 6 shows an equity curve for this stochastic system trading one contract per trade of the S&P 500 full-sized futures contract from 1982 to 2014 using data from Pinnacle Data Corp. Slippage & commission of $100 per round-turn trade were subtracted from each trade.

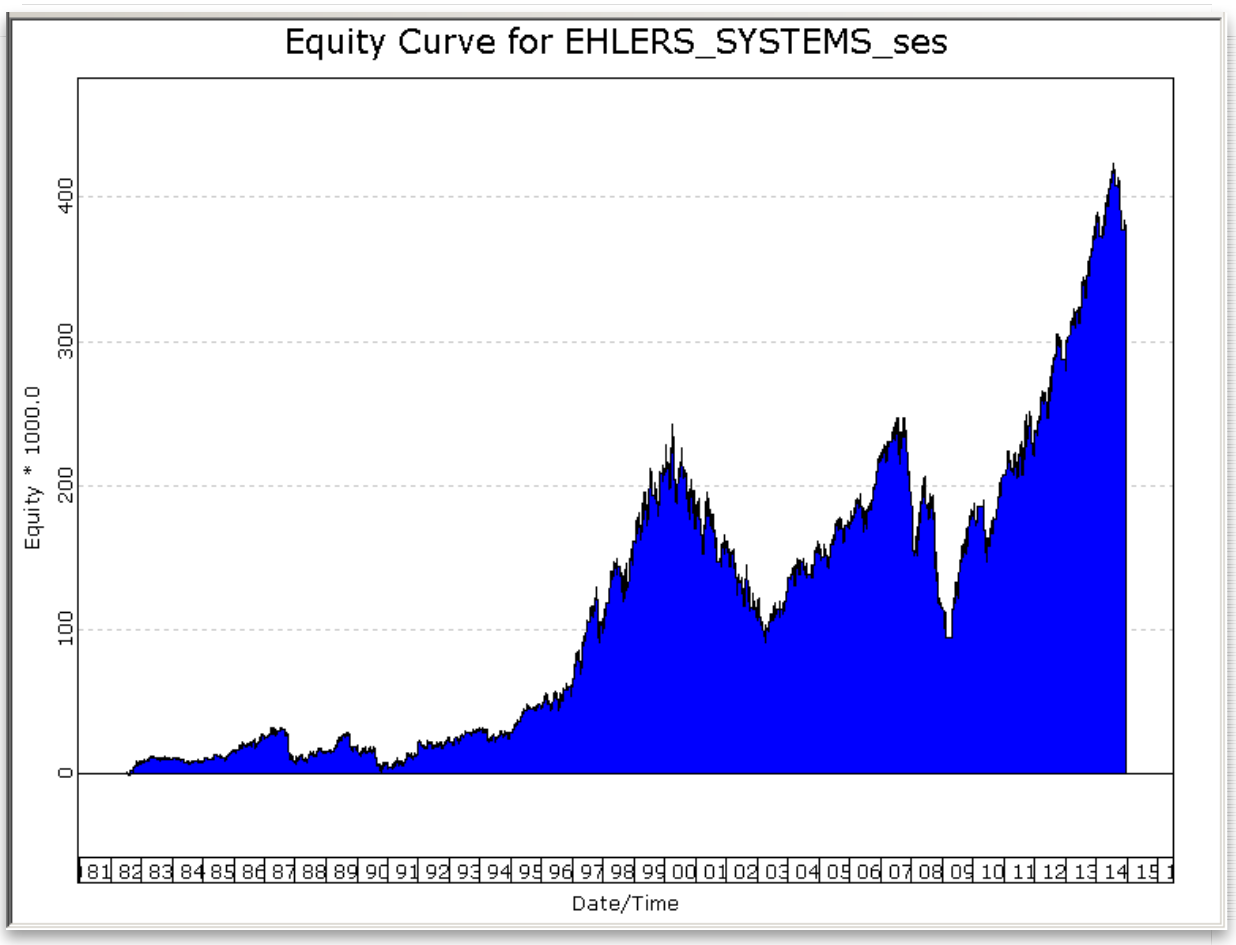


FIGURE 6: TRADERSSTUDIO. Here is a sample equity curve trading the stochastic system one contract per trade of the S&P 500 full-sized futures contract from 1982 to 2014.

```
'TRADING SYSTEM DESIGN: A STATISTICAL APPROACH

'Author: John Ehlers, TASC March 2015

'Coded by: Richard Denning 1/12/2015

'www.TradersEdgeSystems.com


'Stochastic trading system from article:

Sub EHLERS_SYSTEMS(StocLength, Threshold, TradeLength, PctLoss)

    Dim HiC As BarArray

    Dim LoC As BarArray

    Dim Stoc As BarArray


    If BarNumber=FirstBar Then

        'StocLength = 8

        'Threshold = .3

        'TradeLength = 14

        'PctLoss = 3.8

        HiC = 0

        LoC = 0

        Stoc = 0

    End If


    HiC = Highest(Close, StocLength, 0)

    LoC = Lowest(Close, StocLength, 0)

    Stoc = (Close - LoC) / (HiC - LoC)


    If CrossesUnder(Stoc, Threshold) Then
```

```
        Buy("LE", 1, 0, Market, Day)

    End If

    If BarsSinceEntry -1>= TradeLength Then

        ExitLong("LX_time", "", 1, 0, Market, Day)

    End If

    If Low < EntryPrice*(1 - PctLoss /100) Then

        ExitLong("LX_loss", "", 1, 0, Market, Day)

    End If

End Sub
```

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

NINJATRADER: MARCH 2015

The SimpleStochastic strategy presented in John Ehlers & Ric Way's article in this issue, "Trading System Design: A Statistical Approach," has been made available for download at www.ninjatrader.com/SC/March2015SC.zip.

Once it has been downloaded, from within the NinjaTrader Control Center window, select the menu File → Utilities → Import NinjaScript and select the downloaded file. This file is for NinjaTrader version 7 or greater.

You can review the strategy source code by selecting the menu Tools → Edit NinjaScript → Strategy from within the NinjaTrader Control Center window and selecting the SimpleStochastic file.

NinjaScript uses compiled DLLs that run native, not interpreted, which provides you with the highest performance possible.

A sample chart implementing the strategy is shown in Figure 7.

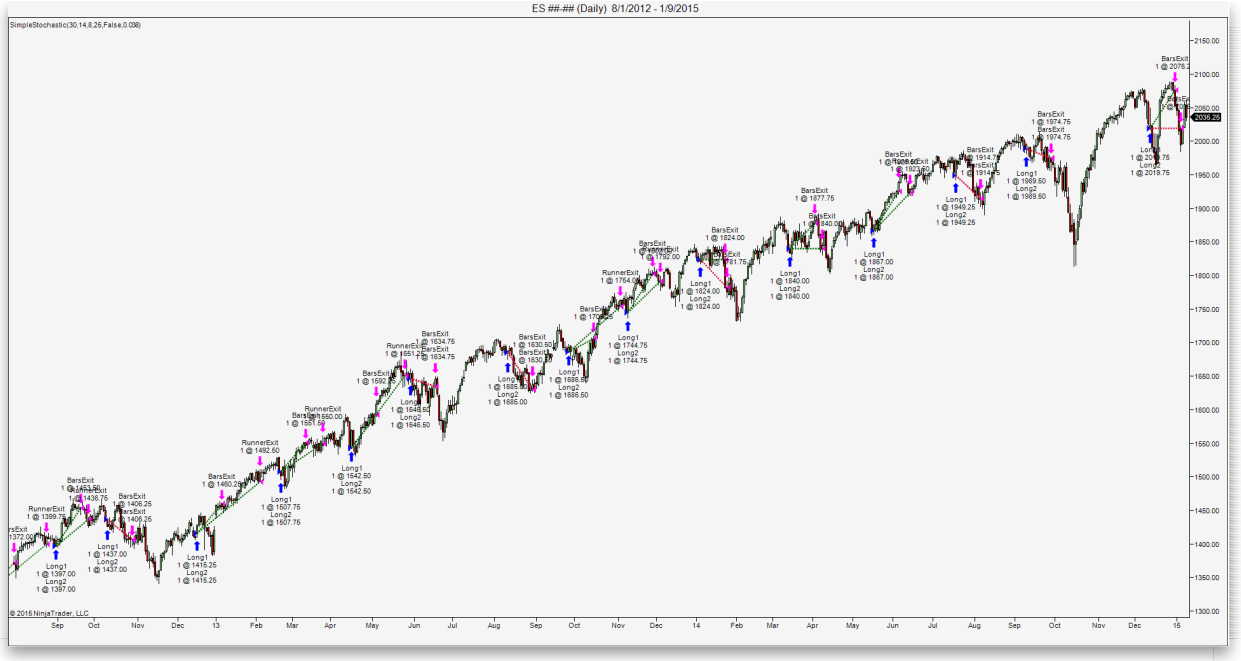
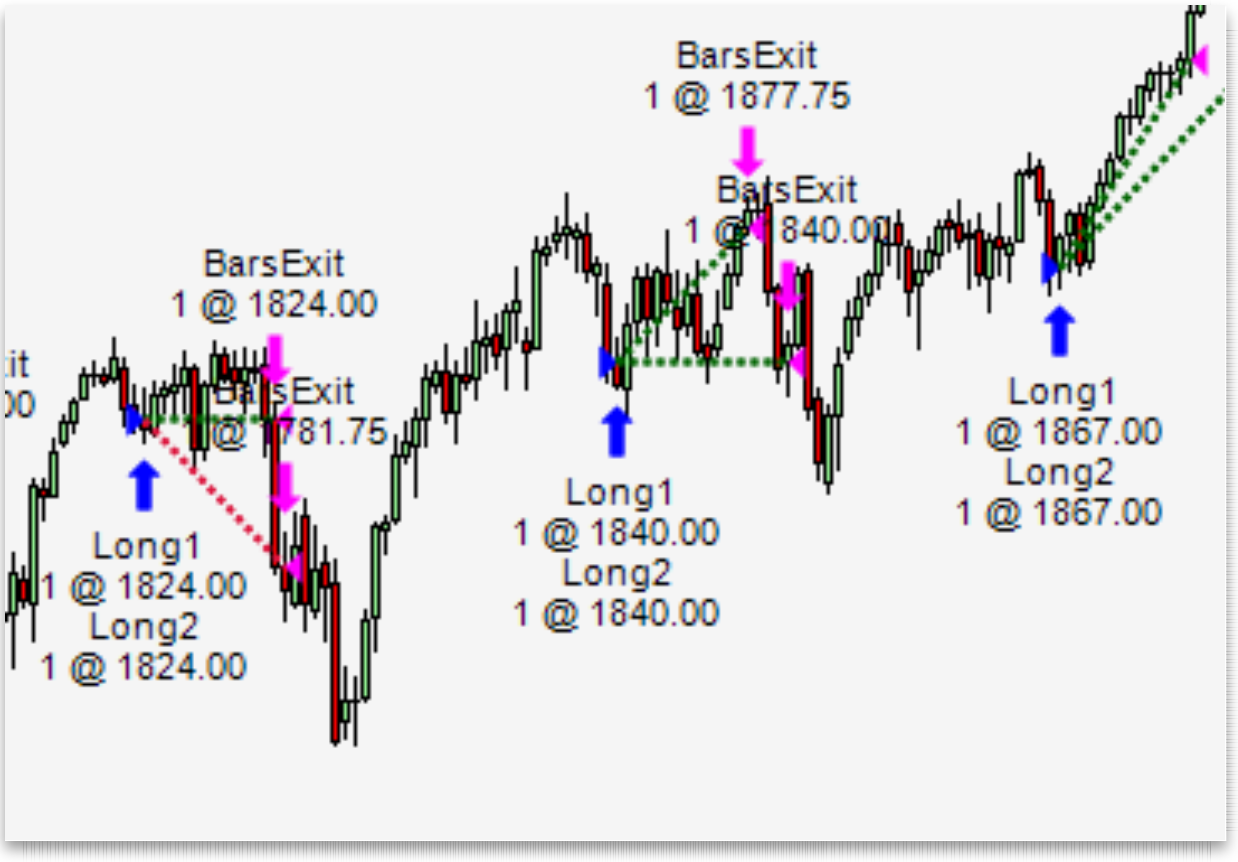


FIGURE 7: NINJATRADER. This screenshot shows the SimpleStochastic strategy applied to a daily emini S&P futures continuous chart in NinjaTrader.



DETAIL FROM FIGURE 7.

—Raymond Deux and Dave Ingram

UPDATA: MARCH 2015

Our Traders' Tip for this month is based on the article by John Ehlers & Ric Way in this issue, "Trading System Design: A Statistical Approach." In it, the authors develop a statistical methodology for the predictability of an event—in this case, the crossing of a stochastic threshold level. By offsetting entry times and measuring the effect this has on overall profitability in the intervening period, a probability distribution function can be created.

The Updata code based on the article is in the Updata Library and may be downloaded by clicking the *custom* menu and *system library*. Those who cannot access the library due to firewall issues may paste the code shown here into the Updata custom editor and save it.

```
'AStochasticSystem
DISPLAYSTYLE 2LINES
INDICATORTYPE TOOL
COLOUR RGB(0,0,200)
COLOUR2 RGB(0,0,200)
PARAMETER "Stochastic Period" #STOCHPERIOD=14
PARAMETER "Threshold" @THRESHOLD=0.3
PARAMETER "Hold Period" #HOLDPERIOD=14
PARAMETER "Stop Loss %" @STOP=3.8
NAME "STOCHASTIC SYSTEM [" #STOCHPERIOD "|" @THRESHOLD "|" #HOLDPERIOD "|" @STOP "]"
""
@UPPER=0
@LOWER=0
@STOCH=0
@ENTRYPRICE=0
FOR #CURDATE=#STOCHPERIOD TO #LASTDATE
   @UPPER=PHIGH(CLOSE,#STOCHPERIOD)
   @LOWER=PLOW(CLOSE,#STOCHPERIOD)
   @STOCH=(CLOSE-@LOWER)/(@UPPER-@LOWER)
   'STOCHASTIC ENTRY
   IF HIST(@STOCH<@THRESHOLD,1) AND ORDERISOPEN=0
      BUY OPEN
      @ENTRYPRICE=OPEN
   ENDIF
   'TIME EXIT
   IF ORDEROPENFOR>=#HOLDPERIOD
      SELL CLOSE
   ENDIF
   '% STOP EXIT
   IF HIST(LOW<@ENTRYPRICE*(1-(@STOP/100)),1)
      SELL OPEN
```

```
    ENDIF
    @PLOT=@UPPER
    @PLOT2=@LOWER
NEXT
```

A sample chart implemention is shown in Figure 8.

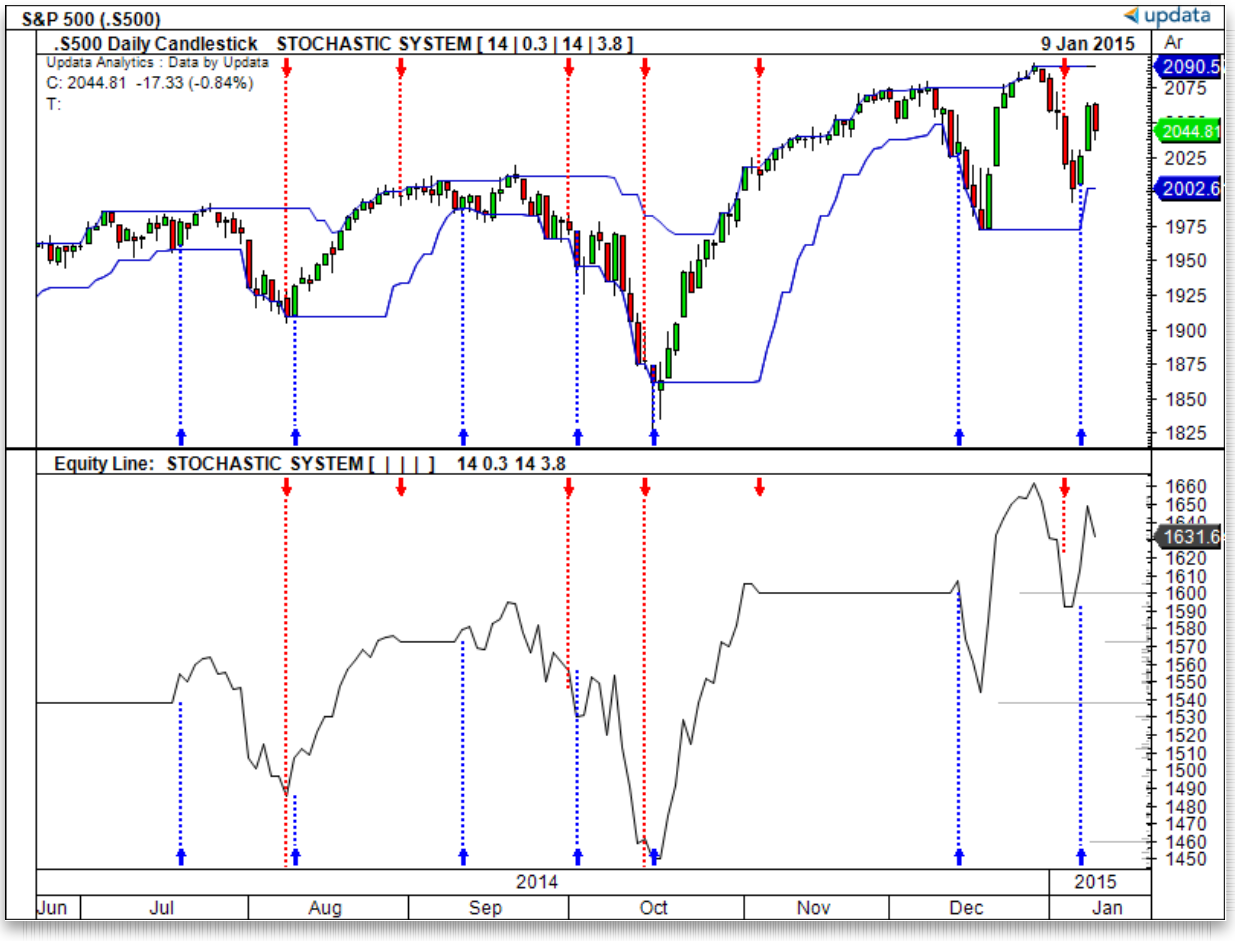


FIGURE 8: UPDATA. Here is an example chart of the simple stochastic entry system as applied to the cash S&P 500 index.

—Updata support team
support@updata.co.uk
www.updata.co.uk

BACK TO LIST

AMIBROKER: MARCH 2015

In “Trading System Design: A Statistical Approach” in this issue, authors John Ehlers & Ric Way

present a way to find out whether signals generated by a given indicator have a statistical edge.

Listing 1 presents AmiBroker Formula Language (AFL) code that produces a profitability distribution chart for a simple statistic crossover system. One can replace the *event* variable with any other system to test its statistical edge. When code is used in AmiBroker's exploration mode, it produces an extra tab(s) with a profitability distribution chart for each symbol separately. To use the formula, type the code into the formula editor and press *send to analysis* to perform an exploration. As you can see from Figure 9, using more data (in this case, hourly) produces a smoother chart than what was presented in the article.

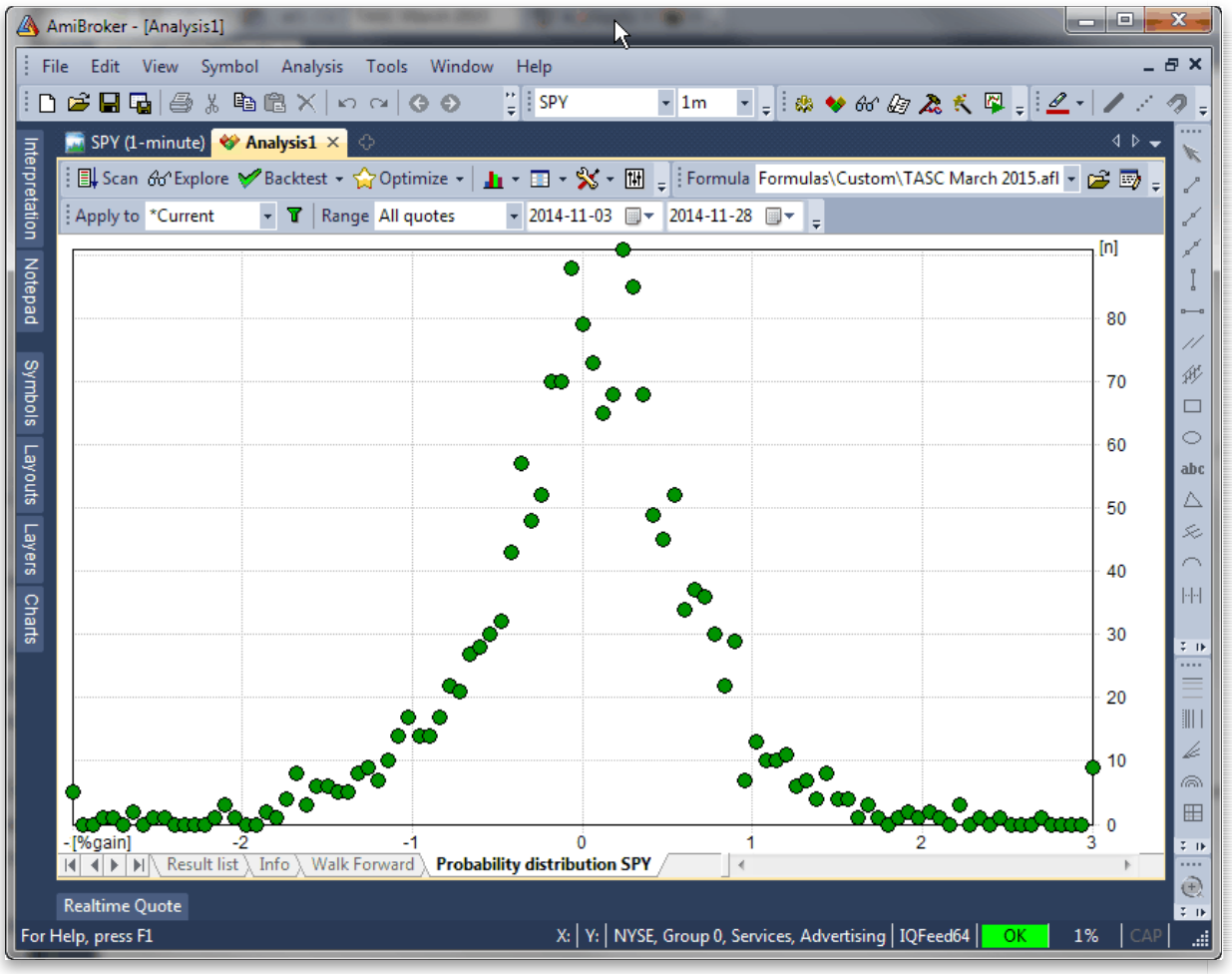


FIGURE 9: AMIBROKER. Here is an AmiBroker exploration chart showing a sample profitability distribution for the stochastic indicator crossing under 0.2 using hourly SPY data. (Note that hourly data and a significantly larger dataset produces a distribution that more closely resembles a classic bell curve than the chart that was shown in Ehlers & Way's article.

LISTING 1.

```
Range = 10;

HiC = HHV( Close, Range );
```

```
LoC = LLV( Close, Range );
Stoc = ( Close - LoC ) / ( HiC - LoC );

Lookback = Range - 1;

Event = Ref( Cross( Stoc, 0.2 ), -Lookback );

PctGainRange = 3; // defines % gain range for X axis

FuturePrice = ROC( Close, Lookback );
// keep values in range
FuturePrice = Min( PctGainRange, Max( -PctGainRange, FuturePrice ) );
// map range to to 0..100
FuturePrice = Round( 100 * ( FuturePrice + PctGainRange )/
              ( 2 * PctGainRange ) );

PredictBin = 0;

for( i = 0; i < BarCount AND BarCount > 100; i++ )
{
    if( Event[ i ] ) PredictBin[ FuturePrice[ i ] ]++;
}

chartname = "Probability distribution " + Name();

XYChartSetAxis(chartname, "[%gain]", "[n]" );
for( i = 0; i < BarCount AND i <= 100; i++ )
{
    XYChartAddPoint( chartname, "",
                     ( i * 2 * PctGainRange / 100 - PctGainRange ),
                     PredictBin[ i ], colorGreen );
}
```

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

BACK TO LIST

MICROSOFT EXCEL: MARCH 2015

In their article in this issue, “Trading System Design: A Statistical Approach,” authors John Ehlers & Ric Way show us a statistical approach to determine if an event we can define to a computer has any value as a future price predictor.

Once we have determined the size and shape of such an event, we can build trading rules around the event and construct a system to follow those rules.

In the article, the authors use a simple stochastic crossunder as the event and look ahead a number of bars to determine a percentage change after the event.

Run this logic against 10 or more years of historical data, accumulate the events you find as well as the percent change values associated with the events, and you can then use a center of gravity (weighted average) calculation to assess the predictive power of the event. The premise

here is that the more positive the CG value, the better your event is likely to be for trading long positions.

Figure 10 shows the specification for one such stochastic event definition on the left under the heading “predictive event testing controls.” The corresponding “event count by price gains” chart with a marker for the calculated center of gravity is shown under the price chart.

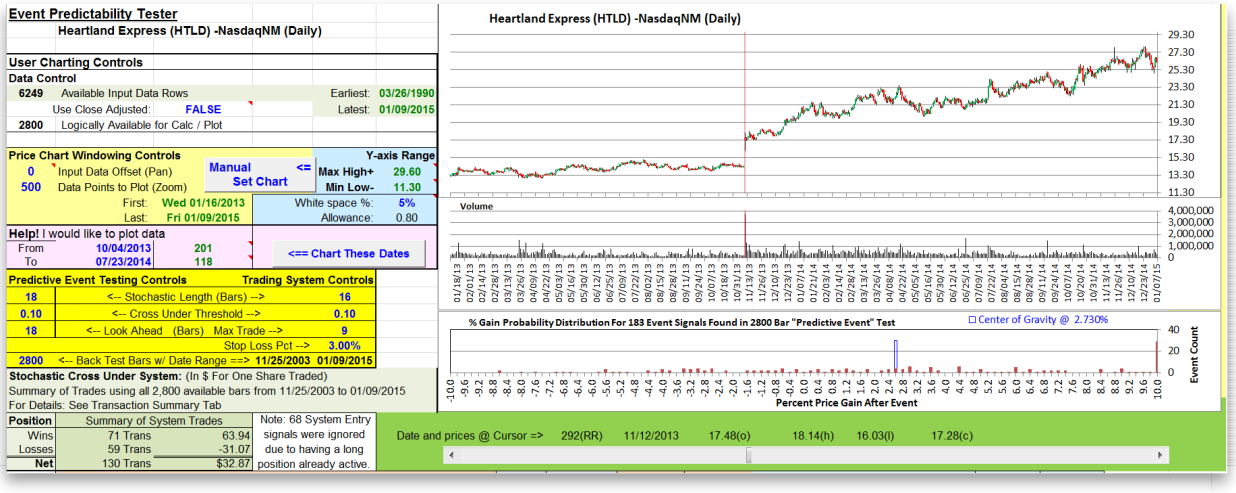
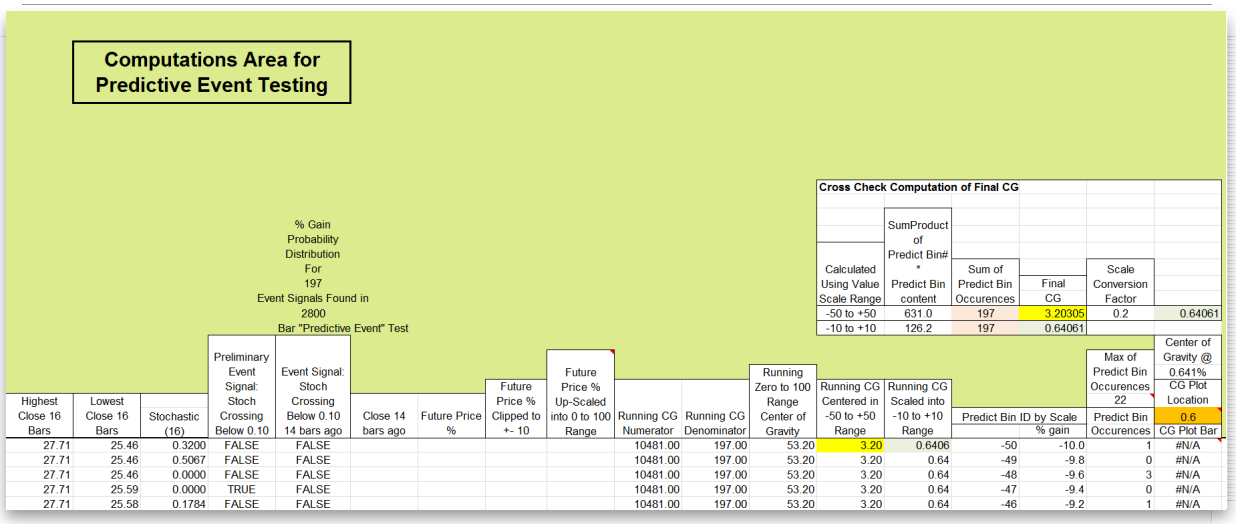


FIGURE 10: EXCEL, Event testing controls and trading controls. This shows the specification for one stochastic event definition on the left under the heading “predictive event testing controls.”

Controls specifying a slightly different size and shape of our event to be used in the simplified trading system appear under the heading “trading system controls.” A summary of the trading results for this control set can be found in the lower-left corner.

Calculations for predictive event testing and center of gravity determination can be found in the columns to the right of the price chart, as shown in Figure 11.



Computations Area for Predictive Event Testing

% Gain Probability Distribution For 197 Event Signals Found in 2800 Bar "Predictive Event" Test

Cross Check Computation of Final CG

Calculated Using Value Scale Range	SumProduct of Predict Bin# * Predict Bin content	Sum of Predict Bin Occurences	Final CG	Scale Conversion Factor	
-50 to +50	631.0	197	3.20305	0.2	0.64061
-10 to +10	126.2	197	0.64061		

Highest Close 16 Bars	Lowest Close 16 Bars	Stochastic (16)	Preliminary Event Signal: Stoch Crossing Below 0.10	Event Signal: Stoch Crossing Below 0.10 14 bars ago	Close 14 bars ago	Future Price %	Future Price % Clipped to +- 10	Future Price % Up-Scaled into 0 to 100 Range	Running CG Numerator	Running CG Denominator	Running Zero to 100 Range Center of Gravity	Running CG Centered in -50 to +50 Range	Running CG Scaled into -10 to +10 Range	Predict Bin ID by Scale	% gain	Max of Predict Bin Occurences 22	Predict Bin Occurences	Center of Gravity @ 0.641% CG Plot Location 0.6 CG Plot Bar
27.71	25.46	0.3200	FALSE	FALSE					10481.00	197.00	53.20	3.20	0.6406	-50	-10.0		1	#N/A
27.71	25.46	0.5067	FALSE	FALSE					10481.00	197.00	53.20	3.20	0.64	-49	-9.8		0	#N/A
27.71	25.46	0.0000	FALSE	FALSE					10481.00	197.00	53.20	3.20	0.64	-48	-9.6		3	#N/A
27.71	25.59	0.0000	TRUE	FALSE					10481.00	197.00	53.20	3.20	0.64	-47	-9.4		0	#N/A
27.71	25.58	0.1784	FALSE	FALSE					10481.00	197.00	53.20	3.20	0.64	-46	-9.2		1	#N/A

FIGURE 11: EXCEL, Predictive Event Computations.

Calculations for predictive event testing and center of gravity determination can be found in the columns to the right of the price chart.

Figure 12 shows the calculations for the trading system. These are located in columns yet farther to the right of those shown in Figure 11.

Computations Area for Trade Decisions

Entries Taken	Entries Ignored Due to Transaction in Progress
122	62

Entry Signal Count
184

Highest Close 18 Bars	Lowest Close 18 Bars	Stochastic (18)	Entry Signal: Stoch Crossing Below 0.10	Stop Loss Delta	Stop Loss Price @ 3.00%	Hit Stop Loss	Trade Duration	Hit Max Trade Duration of 9 Bars	Composite of System Exit Conditions	Transaction Decision Table: Entries and Exits shown here are on the bar AFTER a Signal. **Note:** Long Only System		
										Enter	Active	Exit
27.71	25.46	0.3200	FALSE	0.77	24.87	FALSE	3	FALSE	FALSE	FALSE	TRUE	FALSE
27.71	25.46	0.5067	FALSE	0.77	24.87	FALSE	2	FALSE	FALSE	FALSE	TRUE	FALSE
27.71	25.46	0.0000	FALSE	0.77	24.87	FALSE	1	FALSE	FALSE	TRUE	TRUE	FALSE
27.71	25.50	0.0407	TRUE	0.00	0.00	FALSE	0	FALSE	FALSE	FALSE	FALSE	FALSE

FIGURE 12: EXCEL, Trading decisions. This shows the calculations for the trading system.

As described in the article, selecting the correct combination of specifications for our predictive event can be a trial & error process. In the spreadsheet I am providing, I have included a rudimentary mechanism to assist with the tedious business of evaluating an array of event parameter choices to find the combination that generates the most promising center of gravity value.

Figure 13 shows this mechanism on the PredictiveEventScenarioTester tab of the workbook. Filling in the values in blue defines the envelope of events we want to look at. In this case, we are set up to look at all stochastic lookback lengths from eight to 18 bars; use threshold values from 0.1 to 0.35 in steps of 0.05; and try look-ahead periods from five to 18 bars, inclusive.

Predictive Event Scenario Test Generator

Control Combinations Generator Specifications

	Stochastic Length (Bars)	Cross Under Threshold	Look Ahead (Bars)
From	8	0.10	5
To	18	0.35	18
Step Size	1	0.05	1
# of Steps	11	6	14

The above settings will generate 924 Control Set Combinations

Notes:
Bars valued control specs must be in whole numbers. No fractional bars!
Be careful of how many control set combinations your specifications will generate as the run will take more time as the combination count goes up. That means your time to complete could go from now to a cup of coffee or from a cup of coffee to some time tomorrow

Run Predictive Event Scenarios

Results for Test of Scenarios: Sorted with best CG listed first

Combination Number	Stochastic Length (Bars)	Cross Under Threshold	Look Ahead (Bars)	Event Count	Center of Gravity
854	18	0.10	18	183	2.7301
770	17	0.10	18	191	2.6953
784	17	0.15	18	203	2.5980
769	17	0.10	17	191	2.5979
868	18	0.15	18	200	2.5080
685	16	0.10	17	197	2.4223
783	17	0.15	17	203	2.4148

FIGURE 13: EXCEL, Finding a "Good" Set of Event Parameters. On the PredictiveEventScenarioTester tab of the workbook, I have included a rudimentary mechanism to assist with evaluating an array of event parameter choices to find the combination that generates the most promising center of gravity value.

When you click the *run* button, the VBA code behind the button cycles through all possible combinations of these values one at a time. The results area keeps track of the looping process.

The results are sorted from highest to lowest on the center of gravity value, and the three control values for this "best" setting are used to set the CalculationsAndCharts (Figure 10) *predictive event* controls.

The TradingSystem-Evaluator tab shown in Figure 14 serves the same purpose for evaluating sets of trading system controls. Control values from the best *equity value* row are used to set the CalculationsAndCharts trading system controls.

Trade Scenario Test Generator

Control Combinations Generator Specifications

	Stochastic Length (Bars)	Cross Under Threshold	Max Trade (Bars)	Stop Loss Percent
From	8	0.10	5	2.00%
To	18	0.35	18	3.00%
Step Size	1	0.05	1	0.50%
# of Steps	11	6	14	3

The above settings will generate 2772 Control Set Combinations

Notes:
Control specs valued in bars must be whole numbers. No fractional bars!
Be careful of how many combinations your specifications will generate as the run will take more time as the combination count goes up.

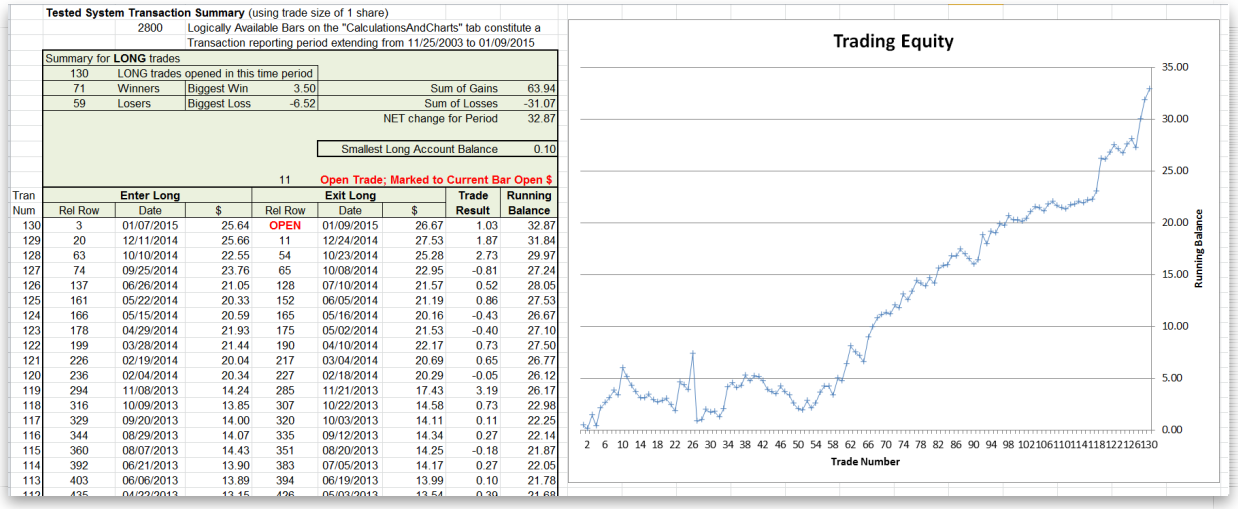
Run Trading System Scenarios

Results for Test of Scenarios: Sorted with best Equity Value listed first

Combination Number	Stochastic Length (Bars)	Cross Under Threshold	Max Trade (Bars)	Stop Loss Percent	Entry Signals	Trades Taken	Final Equity Value
2031	16	0.10	9	3.00%	198	130	$32.87
2283	17	0.10	9	3.00%	192	128	$32.74
18	8	0.10	10	3.00%	291	177	$30.62
2046	16	0.10	14	3.00%	198	119	$30.29
17	8	0.10	10	2.50%	291	190	$30.12
2281	17	0.10	9	2.00%	192	127	$29.62

FIGURE 14: EXCEL, Finding a "Good" Set of Event Parameters (Cont'd.). The TradingSystemEvaluator tab serves the same purpose for evaluating sets of trading system controls. Here, control values from the best equity value row are used to set the CalculationsAndCharts trading system controls.

Trading results for a given scenario can be seen on the *transaction summary* tab shown in Figure 15.



Tested System Transaction Summary (using trade size of 1 share)

2800 Logically Available Bars on the "CalculationsAndCharts" tab constitute a Transaction reporting period extending from 11/25/2003 to 01/09/2015

Summary for **LONG** trades					
130	LONG trades opened in this time period				
71	Winners	Biggest Win	3.50	Sum of Gains	63.94
59	Losers	Biggest Loss	-6.52	Sum of Losses	-31.07
				NET change for Period	32.87
				Smallest Long Account Balance	0.10
			11	Open Trade; Marked to Current Bar Open $	

Tran	Enter Long			Exit Long			Trade	Running
Num	Rel Row	Date	$	Rel Row	Date	$	Result	Balance
130	3	01/07/2015	25.64	OPEN	01/09/2015	26.67	1.03	32.87
129	20	12/11/2014	25.66	11	12/24/2014	27.53	1.87	31.84
128	63	10/10/2014	22.55	54	10/23/2014	25.28	2.73	29.97
127	74	09/25/2014	23.76	65	10/08/2014	22.95	-0.81	27.24
126	137	06/26/2014	21.05	128	07/10/2014	21.57	0.52	28.05
125	161	05/22/2014	20.33	152	06/05/2014	21.19	0.86	27.53
124	166	05/15/2014	20.59	165	05/16/2014	20.16	-0.43	26.67
123	178	04/29/2014	21.93	175	05/02/2014	21.53	-0.40	27.10
122	199	03/28/2014	21.44	190	04/10/2014	22.17	0.73	27.50
121	226	02/19/2014	20.04	217	03/04/2014	20.69	0.65	26.77
120	236	02/04/2014	20.34	227	02/18/2014	20.29	-0.05	26.12
119	294	11/08/2013	14.24	285	11/21/2013	17.43	3.19	26.17
118	316	10/09/2013	13.85	307	10/22/2013	14.58	0.73	22.98
117	329	09/20/2013	14.00	320	10/03/2013	14.11	0.11	22.25
116	344	08/29/2013	14.07	335	09/12/2013	14.34	0.27	22.14
115	360	08/07/2013	14.43	351	08/20/2013	14.25	-0.18	21.87
114	392	06/21/2013	13.90	383	07/05/2013	14.17	0.27	22.05
113	403	06/06/2013	13.89	394	06/19/2013	13.99	0.10	21.78

FIGURE 15: EXCEL, Trade Details. Trading results for a given scenario can be seen on the transaction summary tab.

You will find that the automated event evaluator and the trading evaluator usually come up with different event specifications as being their best choice. I think one of the reasons for this difference can be found in the trading summary in the bottom left of Figure 10: Not every event/entry signal participates in a trade. Many are ignored because a trade is already in progress. So while each of these "ignored" events contributed to a center of gravity computation, they do not contribute independently to the equity value. Moreover, stop-loss processing of a trade may prevent an event entry from reaching the potential contribution that was recognized in the event evaluator CG calculation.

The spreadsheet file for this Traders' Tip (EventPredictabilityTester.xlsm) can be downloaded here. To successfully download it, follow these steps:

- Right-click on the Excel file link, then
- Select "save as" (or "save target as") to place a copy of the spreadsheet file on your hard drive.

—Ron McAllister
Excel and VBA programmer
rpmac_xltt@sprynet.com

BACK TO LIST

Originally published in the March 2015 issue of
Technical Analysis of STOCKS & COMMODITIES magazine.

April 2015

For this month's Traders' Tips, the focus is mainly Dave Cline's article in this issue, "Basket Trading Using A Directed Acyclic Graph", w ith some Traders' Tips focusing on "Kiss & Touch With The Modified True Range" by Chris Lindgren, w hich appeared in the February 2015 issue. Here, w e present the April 2015 Traders' Tips code w ith possible implementations in various softw are.

Code already provided by Cline or Lindgren in their articles can be found in the Subscriber Area of our w ebsite here.

The Traders' Tips section is provided to help the reader implement a selected technique from an article in this issue or another recent issue. The entries here are contributed by softw are developers or programmers for softw are that is capable of customization.

TRADESTATION: APRIL 2015

In the February 2015 STOCKS & COMMODITIES article "Kiss & Touch With The Modified True Range," author Chris Lindgren describes an indicator to help estimate the probability of an option's underlying touching or reaching a certain price any time before expiration. Lindgren describes several w ays that he uses the new indicator, w hich is itself is a modification to J. Welles Wilder's *true range*.

For users' convenience, w e are providing EasyLanguage code for implementing the indicator in TradeStation.

Indicator: Modified True Range

```
inputs:
        Goal( 10 ),
        PeriodLength( 40 ) ;

variables:
        MTR( 0 ),
        AvgMTR( 0 ),
        StdDevMTR( 0 ),
        GoalCount( 0 ),
        GoalPctAchieved( 0 ),
        GoalAchieved( 0 ) ;
```

```
MTR = Maxlist( AbsValue( High - Close[1] ),
        AbsValue( Low - Close[1] ) ) ;

AvgMTR = Average( MTR, PeriodLength ) ;

StdDevMTR = StdDev( MTR, PeriodLength ) ;

if  MTR >= Goal then
        begin
        GoalAchieved = 1 ;
        SetPlotColor( 10, Green ) ;
        end
else
        begin
        GoalAchieved = 0 ;
        SetPlotColor( 10, Red ) ;
        end ;

GoalCount = CountIf( GoalAchieved = 1, PeriodLength ) ;
GoalPctAchieved = iff( PeriodLength > 0, 100
        * GoalCount / PeriodLength, 0 ) ;

//Status Line Plots
Plot1( "Goal Count : " + NumToStr ( GoalCount, 0 )
        + Spaces( 5 ) , "Count", Yellow ) ;
Plot2( "Goal PCT : " + NumToStr ( GoalPctAchieved, 2 )
        + "%" + Spaces( 5 ) , "CountPct", Yellow ) ;
Plot3( "Avg MTR : " + NumToStr ( AvgMTR, 2 )
        + Spaces( 5 ) , "AVGMTR", Yellow ) ;
Plot4( "MTR Std Dev : " + NumToStr ( StdDevMTR, 2 )
        + Spaces( 5 ) , "MTR Std Dev", Yellow ) ;

//Chart Plots
Plot10( MTR, "MTR" ) ;
Plot11( Goal, "Goal" ) ;
```

To dow nload the EasyLanguage code, please visit our TradeStation and EasyLanguage support forum. The code from this article can be found here: http://w w w .tradestation.com/TASC-2015. The ELD filename is "_TASC_ModifiedTrueRange.ELD." For more information about EasyLanguage in general, please see http://w w w .tradestation.com/EL-FAQ.

A sample chart is show n in Figure 1.

FIGURE 1: TRADESTATION, MODIFIED TRUE RANGE. Here is an example of the modified true range indicator applied to a daily chart of Google (GOOG).

This article is for informational purposes. No type of trading or investment recommendation, advice, or strategy is being made, given, or in any manner provided by TradeStation Securities or its affiliates.

—Doug McCrary
TradeStation Securities, Inc.
www.TradeStation.com

BACK TO LIST

eSIGNAL: APRIL 2015

For this month's Traders' Tip we've provided the formula ModifiedTrueRange.efs based on the formula

described in Chris Lindgren's article in the February 2015 issue of S&C, "Kiss & Touch With The Modified True Range."

The study contains formula parameters to set the desired period and price, w hich may be configured through the *edit chart* w indow (right-click on the chart and select *edit chart*) to set the desired period and price. A sample chart is show n in Figure 2.

FIGURE 2: eSIGNAL, MODIFIED TRUE RANGE. Here is an example of the study on a chart of Google.

To discuss this study or dow nload a complete copy of the formula code, please visit the EFS library discussion board forum under the *forums* link from the support menu at w w w .esignal.com or visit our EFS Know ledgeBase at http://w w w .esignal.com/support/kb/efs/. The eSignal formula script (EFS) is also available for copying & pasting here:

```
/**********************************
Provided By:
    Interactive Data Corporation (Copyright © 2015)
    All rights reserved. This sample eSignal Formula Script (EFS)
    is for educational purposes only. Interactive Data Corporation
    reserves the right to modify and overwrite this EFS file with
    each new release.

Description:
    Kiss & Touch With The Modified True Range by Chris Lindgren
```

```
Version:            1.00  02/06/2015

Formula Parameters:                     Default:
Period                                  20
Grouping Offset                         1
Goal                                    9.99

Notes:
    The related article is copyrighted material. If you are not a subscriber
    of Stocks & Commodities, please visit www.traders.com.

**********************************/

var fpArray = new Array();


function preMain(){


    setStudyTitle("ModifiedTrueRange");


    setCursorLabelName("GoalLevel", 0);
    setCursorLabelName("MTR", 1);

    setDefaultBarFgColor(Color.yellow, 0);
    setDefaultBarFgColor(Color.grey, 1);

    setPlotType(PLOTTYPE_LINE, 0);
    setPlotType(PLOTTYPE_HISTOGRAM, 1);

    setDefaultBarThickness(1, 0);
    setDefaultBarThickness(3, 1);

    var x = 0;

    fpArray[x] = new FunctionParameter("fpPeriod", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("Period");
        setLowerLimit(1);
        setDefault(20);
    }

    fpArray[x] = new FunctionParameter("fpGroupOffset", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("Grouping Offset");
        setLowerLimit(1);
        setDefault(1);
    }

    fpArray[x] = new FunctionParameter("fpGoal", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("Goal");
        setLowerLimit(0);
        setDefault(9.99);
    }
}

var bInit = false;
var bVersion = null;
```

```
var xClose = null;
var xLowest = null;
var xHighest = null;

var xRanges = null;



var xMTR = null;

var xRange = null;



var xHitGoal = null;

var xAvgMTR = null;
var xMTRStdDev = null;



var aLabels = [];

var nButtonSpace = 5;


function main(fpPeriod, fpGroupOffset, fpGoal){


    if (bVersion == null) bVersion = verify();
    if (bVersion == false) return;

    if (!bInit){
        xClose = close();
        xLowest = lowest(fpGroupOffset, low());

        xHighest = highest(fpGroupOffset, high());

        xRanges = efsInternal("Calc_Ranges", xClose, xLowest, xHighest,
fpGroupOffset);



        xMTR = getSeries(xRanges, 0);

        xRange = getSeries(xRanges, 1);



        xHitGoal = efsInternal("Calc_HitGoal", xMTR, fpGoal);



        xAvgMTR = sma(fpPeriod, xMTR);

        xMTRStdDev = efsInternal("Calc_StdDev", xRange, fpPeriod);


        bInit = true;
```

```
    }

    var nMTR = xMTR.getValue(0);

    if (nMTR == null)

        return;

    setBarFgColor((nMTR >= fpGoal) ? Color.green : Color.red, 1);



    var GoalCount = null;

    for (var i = 0; i < fpPeriod; i++){

        var nHitGoal = xHitGoal.getValue(-i);

        if (nHitGoal == null){

            GoalCount = "n/a"

            break;

        }

        GoalCount += nHitGoal;

    }



    var GoalPct = (GoalCount == "n/a") ? "n/a" : "%" + Math.round(GoalCount /
fpPeriod * 100);



    var AvgMTR = xAvgMTR.getValue(0);

    var MTRStdDev = xMTRStdDev.getValue(0);



    AvgMTR = (AvgMTR == null) ? "n/a" : "$" + AvgMTR.toFixed(2);

    MTRStdDev = (MTRStdDev == null) ? "n/a" : "$" + MTRStdDev.toFixed(2);



    if (isLastBarOnChart()){

        var x = 10;

        var y = 25;



        aLabels = [];

        aLabels.push((fpGroupOffset > 1) ?
```

```
        ["** Special Grouping is set is for " + fpGroupOffset + " bars **", 285,
Color.RGB(255,155,0)] : null);

        aLabels.push(["Period Length " + fpPeriod, 135, Color.lime]);

        aLabels.push(["Goal $" + fpGoal.toFixed(2), 105, Color.lime]);

        aLabels.push(["Goal Count " + GoalCount, 115, Color.lime]);

        aLabels.push(["Percent Goal Achieved " + GoalPct, 180, Color.lime]);

        aLabels.push(["AverageMTR(" + fpPeriod + ") = " + AvgMTR, 215,
Color.lightgrey]);

        aLabels.push(["MTR Std Dev(" + fpPeriod + ") = " + MTRStdDev, 215,
Color.lightgrey]);



        aLabels.forEach(

            function(aValue, nIndex){

                if (aValue != null){

                    drawTextPixel(x, y, aValue[0], Color.navy, aValue[2],

                    Text.BOLD|Text.FRAME|Text.CENTER, "Arial", 8, aValue[0],
aValue[1]);

                    x = x + aValue[1] + nButtonSpace;

                }

                if (nIndex == 4){

                    x = 10;

                    y = 45;

                }

            }

        )

    }


    return [fpGoal, nMTR];
}


function Calc_Ranges(xClose, xLowest, xHighest, nGroupOffset){



    nPrevClose = xClose.getValue(-nGroupOffset);
```

```
    nLowest = xLowest.getValue(0);

    nHighest = xHighest.getValue(0);


    if (nPrevClose == null || nLowest == null || nHighest == null)

        return;


    var nRangeLow = Math.abs(nPrevClose - nLowest);

    var nRangeHigh = Math.abs(nPrevClose - nHighest);


    var nMTR = Math.max(nRangeLow, nRangeHigh);


    var nRangeLow1 = (nPrevClose - nLowest);

    var nRangeHigh1 = (nPrevClose - nHighest);


    var nRange = Math.abs(nRangeLow1) >= Math.abs(nRangeHigh1) ? nRangeLow1 :
nRangeHigh1;


    return [nMTR, nRange];

}


function Calc_StdDev(xSource, nPeriod){


    var nSumX = 0;

    var nSumX2 = 0;


    for (i = 0; i < nPeriod; i++){

        var nSource = xSource.getValue(-i);

        if (nSource == null)

            return;

        nSumX += nSource;

        nSumX2 += (nSource * nSource);
```

```
    }

    var nSMA = (nSumX / nPeriod);
    var nStdDev = Math.sqrt((nSumX2 / nPeriod) - (nSMA * nSMA));

    return nStdDev;
}

function Calc_HitGoal(xMTR, nGoal){

    var nMTR = xMTR.getValue(0);

    if (nMTR == null)
        return;

    var nHitGoal = (nMTR >= nGoal) ? 1 : 0;

    return nHitGoal;
}

function verify(){

    var b = false;
    if (getBuildNumber() < 779){
        drawTextAbsolute(5, 35, "This study requires version 8.0 or later.",
            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,
            null, 13, "error");
        drawTextAbsolute(5, 20, "Click HERE to upgrade.@URL=http://www.esignal.com
/download/default.asp",
            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,
            null, 13, "upgrade");
        return b;
    }
    else
        b = true;

    return b;
}
```

—Eric Lippert
eSignal, an Interactive Data company
800 779-6555, www.eSignal.com

BACK TO LIST

WEALTH-LAB: APRIL 2015

In his February 2015 S&C article, “Kiss & Touch With The Modified True Range,” author Chris Lindgren explored an option’s probability of reaching a certain price prior to expiration. Figuring the probability is achieved through some simple statistical calculations. At the foundation of his work is a new rendition of J. Welles Wilder’s classic indicator *true range* (TR) referred to in Lingren’s articles as the *modified true range* (MTR). The difference between the true range and the modified true range is simply the omission of the current high less the current low. Lindgren’s new indicator only considers the greatest of the following two values:

- Absolute value of the current high less the previous close
- Absolute value of the current low less the previous close

It’s noticeable how the rounding performed by the average MTR indicator makes the result appear smoother than the traditional average true range (ATR).

While option trading is an interesting topic of its own, here we will explore applications of the indicator introduced by Lindgren to trading securities. One example of a possible trading idea could be detection of price spikes using the new MTR indicator. Our sample trading system will follow a countertrend, mean-reversion approach, fading strong moves over a short-term horizon. To spot outlier values in the price, we calculate the standard deviation of the MTR. The setup is in place when the current MTR reading exceeds the average MTR by three standard deviations of MTR or greater. An entry is triggered when the following criteria are met:

- *Buy* next bar at market if today’s close price has declined
- *Short* next bar at market if today’s close price has increased.

A position is exited with a simple trailing channel exit:

- *Exit long* when today’s low price has broken below the three-day lowest price
- *Cover short* when today’s high price has broken above the three-day highest price.

See Figure 3 for an example chart.

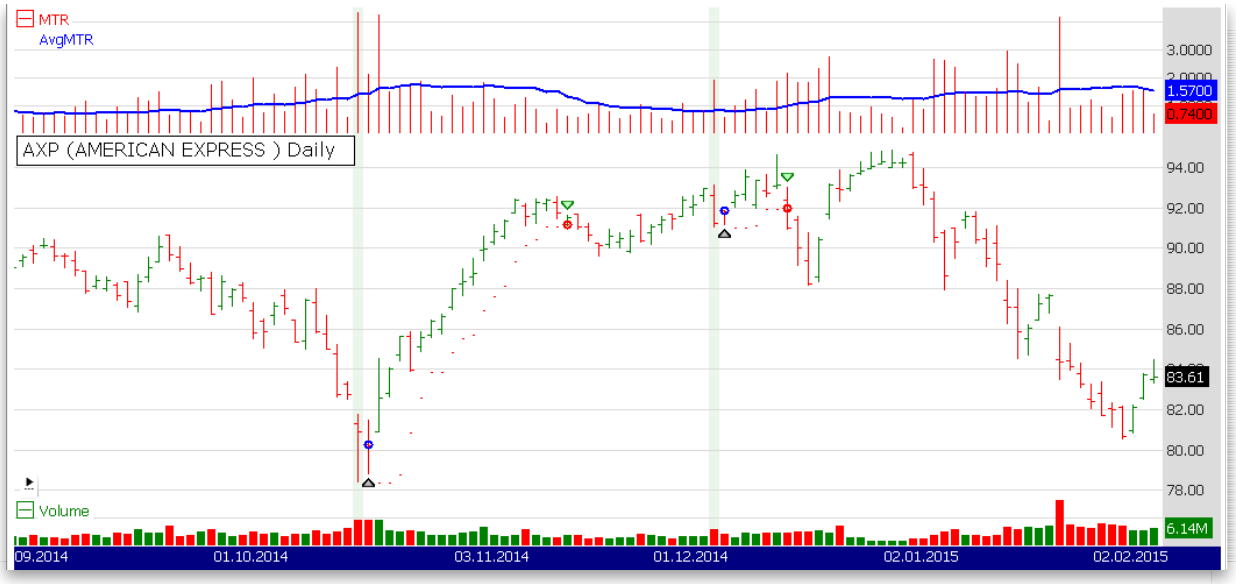


FIGURE 3: WEALTH-LAB, MODIFIED TRUE RANGE. This sample Wealth-Lab 6 chart illustrates the sample system's performance on a daily chart of American Express (AXP).

This system is merely a proof-of-concept. Among possible enhancements to this barebone system, consider the following suggestions:

- Perform trend detection to avoid entering against established trends
- To support the trailing exit, introduce a stop-loss for risk management
- Explore whether "clumping" of MTR spikes has an effect on the system's profitability.

Wealth-Lab 6 strategy code (C#):

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Drawing;
using WealthLab;
using WealthLab.Indicators;

namespace WealthLab.Strategies
{
        public class ModifiedTrueRange : WealthScript
        {
                private StrategyParameter paramPeriod;

                public ModifiedTrueRange()
                {
                        paramPeriod = CreateParameter("Period", 20, 1, 100, 1);
                }

                protected override void Execute()
                {
                        int period = paramPeriod.ValueInt;
                        double Goal = 9.99;

                        //Calcluate modified true range (MTR)
```

```
                        DataSeries MTR = new DataSeries( Bars,"MTR" );
                        DataSeries RangeLow = DataSeries.Abs( (Close>>1)-
Lowest.Series(Low,1) );
                        DataSeries RangeHigh = DataSeries.Abs( (Close>>1)-
Highest.Series(High,1) );
                        DataSeries Range1 = new DataSeries( Bars,"Range" );
                        DataSeries HitGoal = new DataSeries( Bars,"HitGoal" );
                        DataSeries GoalCount = new DataSeries( Bars,"GoalCount" );
                        DataSeries GoalPct = new DataSeries( Bars,"GoalPct" );
                        DataSeries AvgMTR = new DataSeries( Bars,"AvgMTR" );

                        for(int bar = 1; bar < Bars.Count; bar++)
                        {
                                double mtr = Math.Max(RangeLow[bar],RangeHigh[bar]);
                                MTR[bar] = mtr;
                        }

                        //MTR Standard Deviation

                        DataSeries RangeLow1=( (Close>>1) - Lowest.Series(Low,1) );
                        DataSeries RangeHigh1=( (Close>>1) - Highest.Series(High,1) );

                        for(int bar = 1; bar < Bars.Count; bar++)
                        {
                                double range = ( Math.Abs(RangeLow1[bar]) >=
Math.Abs(RangeHigh1[bar]) ) ?
                                        RangeLow1[bar] :
RangeHigh1[bar];
                                Range1[bar] = range;
                        }

                        DataSeries MTRStdDev =
StdDev.Series(Range1,period,StdDevCalculation.Sample);

                        for(int bar = 1; bar < Bars.Count; bar++)
                        {
                                HitGoal[bar] = MTR[bar] >= Goal ? 1 : 0;
                                GoalCount[bar] = Sum.Series( HitGoal,period )[bar];

                                double goalPct = Math.Round( GoalCount[bar]/period,
2) * 100;
                                double avgMTR = Math.Round( SMA.Series(MTR,period)
[bar], 2 );

                                GoalPct[bar] = goalPct;
                                AvgMTR[bar] = avgMTR;
                        }

                        StdDev MTRSD =
StdDev.Series(MTR,period,StdDevCalculation.Sample);

                        ChartPane mtrPane = CreatePane( 40, true, true );
                        PlotSeries( mtrPane, MTR, Color.Red, LineStyle.Histogram, 1 );
                        PlotSeries( mtrPane, AvgMTR, Color.Blue, LineStyle.Solid, 2 );
                        PlotStops();

                        for(int bar = GetTradingLoopStartBar(period); bar <
Bars.Count; bar++)
                        {
                                if (IsLastPositionActive)
```

```
                                {
                                        Position p = LastPosition;
                                        if( p.PositionType == PositionType.Long )
                                                SellAtStop( bar+1, p, Lowest.Series(
Low,3 )[bar] );
                                        else
                                                CoverAtStop( bar+1, p,
Highest.Series( High,3 )[bar] );
                                }
                                else
                                {
                                        //MTR spike
                                        bool spike = MTR[bar] > (AvgMTR[bar] + 3 *
MTRSD[bar]);
                                        if( spike )
                                        {
                                                SetBackgroundColor( bar,
Color.FromArgb( 20, Color.Green ) );
                                                if( ROC.Value( bar, Close, 1 ) > 0 )
                                                        ShortAtMarket(bar+1 );
                                                else
                                                        BuyAtMarket(bar+1 );
                                        }
                                }
                        }
                }
        }
}
```

—Eugene, Wealth-Lab team
MS123, LLC
www.wealth-lab.com

BACK TO LIST

NEUROSHELL TRADER: APRIL 2015

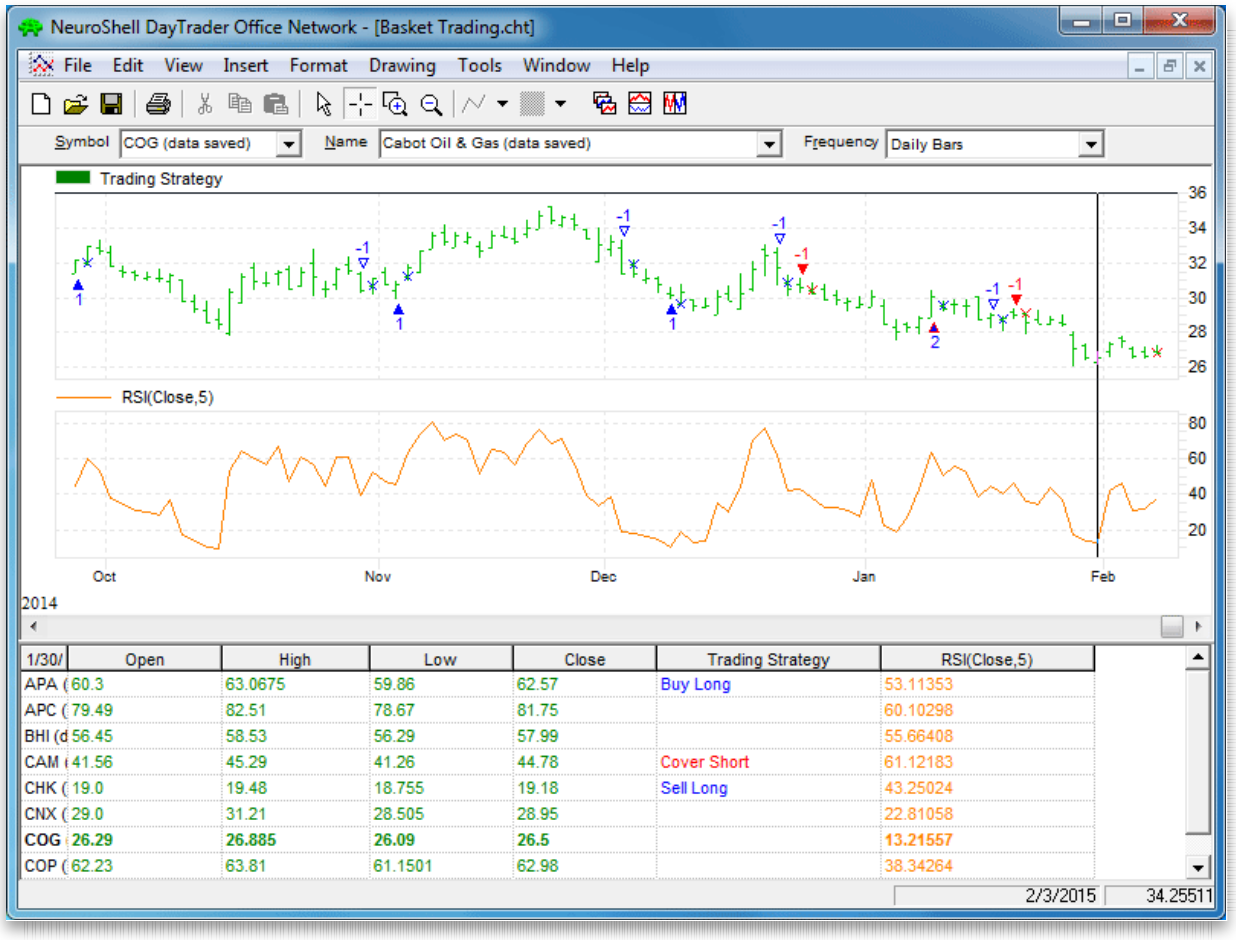
An RSI mean-reverting basket trading system such as the one described in Dave Cline's article in this issue, "Basket Trading Using A Directed Acyclic Graph," can be easily implemented with a few of NeuroShell Trader's 800+ indicators. Simply select *new trading strategy* from the *insert* menu and enter the following in the appropriate locations of NeuroShell's trading strategy wizard:

```
BUY LONG CONDITIONS:            A<=B(ChartPageURank(RSI(Close,5)),3)
SELL LONG CONDITIONS:           A>B(ChartPageURank(RSI(Close,5)),3)
SELL SHORT CONDITIONS:          A<=B(ChartPageLRank(RSI(Close,5)),3)
COVER SHORT CONDITIONS:         A>B(ChartPageLRank(RSI(Close,5)),3)
```

If you have NeuroShell Trader Professional, you can also choose whether the parameters should be optimized. After backtesting the trading strategy, use the *detailed analysis* button to view the backtest and trade-by-trade statistics for the strategy.

Users of NeuroShell Trader can go to the Stocks & Commodities section of the NeuroShell Trader free technical support w ebsite to dow nload a copy of this or any previous Traders' Tips.

A sample chart is show n in Figure 4.



1/30/	Open	High	Low	Close	Trading Strategy	RSI(Close,5)
APA (	60.3	63.0675	59.86	62.57	Buy Long	53.11353
APC (	79.49	82.51	78.67	81.75		60.10298
BHI (d	56.45	58.53	56.29	57.99		55.66408
CAM (	41.56	45.29	41.26	44.78	Cover Short	61.12183
CHK (	19.0	19.48	18.755	19.18	Sell Long	43.25024
CNX (	29.0	31.21	28.505	28.95		22.81058
COG	**26.29**	**26.885**	**26.09**	**26.5**		**13.21557**
COP (	62.23	63.81	61.1501	62.98		38.34264

FIGURE 4: NEUROSHELL TRADER, BASKET TRADING SYSTEM. This NeuroShell Trader chart displays an RSI mean-reverting basket trading system.

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

BACK TO LIST

AIQ: APRIL 2015

The AIQ code based on Dave Cline's article in this issue, "Basket Trading Using A Directed Acyclic Graph," is provided at w w w .TradersEdgeSystems.com/traderstips.htm, and is also show n here:

```
!BASKET TRADING USING A DIRECTED ACYCLIC GRAPH

!Author: Dave Cline, TASC April 2015

!Coded by: Richard Denning 2/10/2015

!www.TradersEdgeSystems.com


!Len is the number of lookback days.

!L1 is the first ticker in the list that you want to compare your list to.

!L2 is the second ticker in the list that you want to compare your list to etc

!You must create a list of 20 or less symbols of interest then type in the same

!symbols in the same alpha order as your list. If there are less than 20 just use " "

!but do not delete the input as there must be L1 through L20 inputs defined:


!INPUTS:

Len is 250.

L1 is "ADBE".

L2 is "ADI".

L3 is "ADP".

L4 is "AKAM".

L5 is "ALTR".

L6 is "AMAT".

L7 is "FOX".

L8 is "INTU".

L9 is "LLTC".

L10 is "LMCA".

L11 is "LRCX".

L12 is "LVNTA".

L13 is "NTAP".

L14 is "NVDA".

L15 is "NXPI".

L16 is "PAYX".
```

```
L17 is "SIRI".

L18 is "TRIP".

L19 is "TXN".

L20 is "VIAB".


!CODE TO CALCULATE PEARSONS r AND r SQUARED:

ChgOC is (([close]-[open])/[open]) * 100.

SumY is Sum(ChgOC, Len).

SumYsq is Sum(ChgOC*ChgOC, Len).

SSy is SumYsq - ( (SumY * SumY) / Len ).


ChgOCL1 is TickerUDF(L1, ChgOC).

SumXsq1 is Sum(ChgOCL1*ChgOCL1, Len).

SumX1 is Sum(ChgOCL1, Len).

SumXY1 is Sum(ChgOC*ChgOCL1, Len).

SP1 is SumXY1 - ( (SumX1 * SumY) / Len ).

SSx1 is SumXsq1 - ( (SumX1 * SumX1) / Len ).

PearR1 is SP1/SQRT(SSx1*SSy).

PearRsq1 is ( PearR1 * PearR1 ).


ChgOCL2 is TickerUDF(L2, ChgOC).

SumXsq2 is Sum(ChgOCL2*ChgOCL2, Len).

SumX2 is Sum(ChgOCL2, Len).

SumXY2 is Sum(ChgOC*ChgOCL2, Len).

SP2 is SumXY2 - ( (SumX2 * SumY) / Len ).

SSx2 is SumXsq2 - ( (SumX2 * SumX2) / Len ).

PearR2 is SP2/SQRT(SSx2*SSy).

PearRsq2 is ( PearR2 * PearR2 ).


ChgOCL3 is TickerUDF(L3, ChgOC).
```

```
SumXsq3 is Sum(ChgOCL3*ChgOCL3, Len).

SumX3 is Sum(ChgOCL3, Len).

SumXY3 is Sum(ChgOC*ChgOCL3, Len).

SP3 is SumXY3 - ( (SumX3 * SumY) / Len ).

SSx3 is SumXsq3 - ( (SumX3 * SumX3) / Len ).

PearR3 is SP3/SQRT(SSx3*SSy).

PearRsq3 is ( PearR3 * PearR3 ).


ChgOCL4 is TickerUDF(L4, ChgOC).

SumXsq4 is Sum(ChgOCL4*ChgOCL4, Len).

SumX4 is Sum(ChgOCL4, Len).

SumXY4 is Sum(ChgOC*ChgOCL4, Len).

SP4 is SumXY4 - ( (SumX4 * SumY) / Len ).

SSx4 is SumXsq4 - ( (SumX4 * SumX4) / Len ).

PearR4 is SP4/SQRT(SSx4*SSy).

PearRsq4 is ( PearR4 * PearR4 ).


ChgOCL5 is TickerUDF(L5, ChgOC).

SumXsq5 is Sum(ChgOCL5*ChgOCL5, Len).

SumX5 is Sum(ChgOCL5, Len).

SumXY5 is Sum(ChgOC*ChgOCL5, Len).

SP5 is SumXY5 - ( (SumX5 * SumY) / Len ).

SSx5 is SumXsq5 - ( (SumX5 * SumX5) / Len ).

PearR5 is SP5/SQRT(SSx5*SSy).

PearRsq5 is ( PearR5 * PearR5 ).


ChgOCL6 is TickerUDF(L6, ChgOC).

SumXsq6 is Sum(ChgOCL6*ChgOCL6, Len).

SumX6 is Sum(ChgOCL6, Len).

SumXY6 is Sum(ChgOC*ChgOCL6, Len).
```

```
SP6 is SumXY6 - ( (SumX6 * SumY) / Len ).

SSx6 is SumXsq6 - ( (SumX6 * SumX6) / Len ).

PearR6 is SP6/SQRT(SSx6*SSy).

PearRsq6 is ( PearR6 * PearR6 ).


ChgOCL7 is TickerUDF(L7, ChgOC).

SumXsq7 is Sum(ChgOCL7*ChgOCL7, Len).

SumX7 is Sum(ChgOCL7, Len).

SumXY7 is Sum(ChgOC*ChgOCL7, Len).

SP7 is SumXY7 - ( (SumX7 * SumY) / Len ).

SSx7 is SumXsq7 - ( (SumX7 * SumX7) / Len ).

PearR7 is SP7/SQRT(SSx7*SSy).

PearRsq7 is ( PearR7 * PearR7 ).


ChgOCL8 is TickerUDF(L8, ChgOC).

SumXsq8 is Sum(ChgOCL8*ChgOCL8, Len).

SumX8 is Sum(ChgOCL8, Len).

SumXY8 is Sum(ChgOC*ChgOCL8, Len).

SP8 is SumXY8 - ( (SumX8 * SumY) / Len ).

SSx8 is SumXsq8 - ( (SumX8 * SumX8) / Len ).

PearR8 is SP8/SQRT(SSx8*SSy).

PearRsq8 is ( PearR8 * PearR8 ).


ChgOCL9 is TickerUDF(L9, ChgOC).

SumXsq9 is Sum(ChgOCL9*ChgOCL9, Len).

SumX9 is Sum(ChgOCL9, Len).

SumXY9 is Sum(ChgOC*ChgOCL9, Len).

SP9 is SumXY9 - ( (SumX9 * SumY) / Len ).

SSx9 is SumXsq9 - ( (SumX9 * SumX9) / Len ).

PearR9 is SP9/SQRT(SSx9*SSy).
```

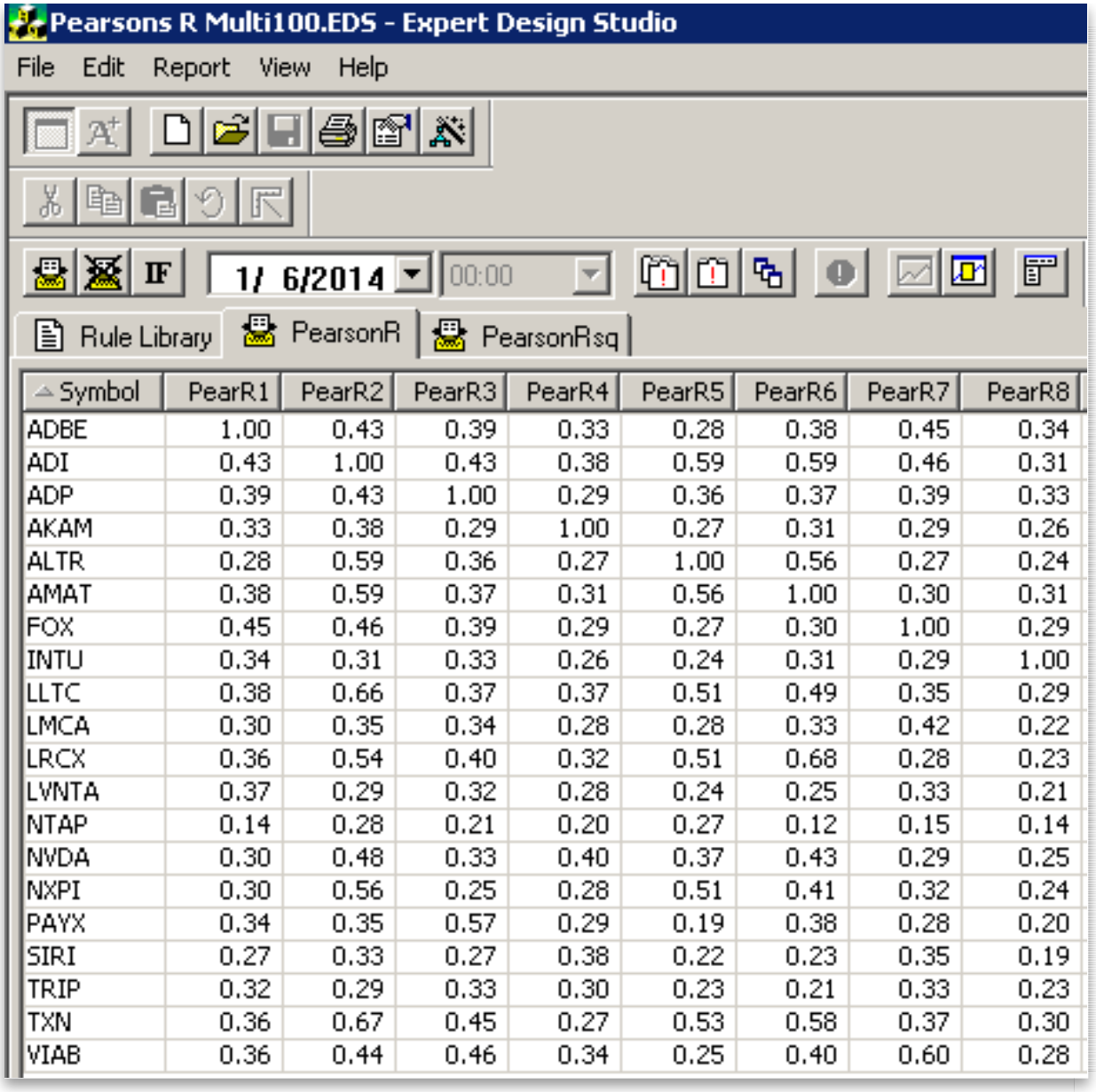
```
PearRsq9 is ( PearR9 * PearR9 ).

ChgOCL10 is TickerUDF(L10, ChgOC).
SumXsq10 is Sum(ChgOCL10*ChgOCL10, Len).
SumX10 is Sum(ChgOCL10, Len).
SumXY10 is Sum(ChgOC*ChgOCL10, Len).
SP10 is SumXY10 - ( (SumX10 * SumY) / Len ).
SSx10 is SumXsq10 - ( (SumX10 * SumX10) / Len ).
PearR10 is SP10/SQRT(SSx10*SSy).
PearRsq10 is ( PearR10 * PearR10 ).

ChgOCL11 is TickerUDF(L11, ChgOC).
SumXsq11 is Sum(ChgOCL11*ChgOCL11, Len).
SumX11 is Sum(ChgOCL11, Len).
SumXY11 is Sum(ChgOC*ChgOCL11, Len).
SP11 is SumXY11 - ( (SumX11 * SumY) / Len ).
SSx11 is SumXsq11 - ( (SumX11 * SumX11) / Len ).
PearR11 is SP11/SQRT(SSx11*SSy).
PearRsq11 is ( PearR11 * PearR11 ).

ChgOCL12 is TickerUDF(L12, ChgOC).
SumXsq12 is Sum(ChgOCL12*ChgOCL12, Len).
SumX12 is Sum(ChgOCL12, Len).
SumXY12 is Sum(ChgOC*ChgOCL12, Len).
SP12 is SumXY12 - ( (SumX12 * SumY) / Len ).
SSx12 is SumXsq12 - ( (SumX12 * SumX12) / Len ).
PearR12 is SP12/SQRT(SSx12*SSy).
PearRsq12 is ( PearR12 * PearR12 ).

ChgOCL13 is TickerUDF(L13, ChgOC).
```

```
SumXsq13 is Sum(ChgOCL13*ChgOCL13, Len).

SumX13 is Sum(ChgOCL13, Len).

SumXY13 is Sum(ChgOC*ChgOCL13, Len).

SP13 is SumXY13 - ( (SumX13 * SumY) / Len ).

SSx13 is SumXsq13 - ( (SumX13 * SumX13) / Len ).

PearR13 is SP13/SQRT(SSx13*SSy).

PearRsq13 is ( PearR13 * PearR13 ).


ChgOCL14 is TickerUDF(L14, ChgOC).

SumXsq14 is Sum(ChgOCL14*ChgOCL14, Len).

SumX14 is Sum(ChgOCL14, Len).

SumXY14 is Sum(ChgOC*ChgOCL14, Len).

SP14 is SumXY14 - ( (SumX14 * SumY) / Len ).

SSx14 is SumXsq14 - ( (SumX14 * SumX14) / Len ).

PearR14 is SP14/SQRT(SSx14*SSy).

PearRsq14 is ( PearR14 * PearR14 ).


ChgOCL15 is TickerUDF(L15, ChgOC).

SumXsq15 is Sum(ChgOCL15*ChgOCL15, Len).

SumX15 is Sum(ChgOCL15, Len).

SumXY15 is Sum(ChgOC*ChgOCL15, Len).

SP15 is SumXY15 - ( (SumX15 * SumY) / Len ).

SSx15 is SumXsq15 - ( (SumX15 * SumX15) / Len ).

PearR15 is SP15/SQRT(SSx15*SSy).

PearRsq15 is ( PearR15 * PearR15 ).


ChgOCL16 is TickerUDF(L16, ChgOC).

SumXsq16 is Sum(ChgOCL16*ChgOCL16, Len).

SumX16 is Sum(ChgOCL16, Len).

SumXY16 is Sum(ChgOC*ChgOCL16, Len).
```

```
SP16 is SumXY16 - ( (SumX16 * SumY) / Len ).

SSx16 is SumXsq16 - ( (SumX16 * SumX16) / Len ).

PearR16 is SP16/SQRT(SSx16*SSy).

PearRsq16 is ( PearR16 * PearR16 ).


ChgOCL17 is TickerUDF(L17, ChgOC).

SumXsq17 is Sum(ChgOCL17*ChgOCL17, Len).

SumX17 is Sum(ChgOCL17, Len).

SumXY17 is Sum(ChgOC*ChgOCL17, Len).

SP17 is SumXY17 - ( (SumX17 * SumY) / Len ).

SSx17 is SumXsq17 - ( (SumX17 * SumX17) / Len ).

PearR17 is SP17/SQRT(SSx17*SSy).

PearRsq17 is ( PearR17 * PearR17 ).


ChgOCL18 is TickerUDF(L18, ChgOC).

SumXsq18 is Sum(ChgOCL18*ChgOCL18, Len).

SumX18 is Sum(ChgOCL18, Len).

SumXY18 is Sum(ChgOC*ChgOCL18, Len).

SP18 is SumXY18 - ( (SumX18 * SumY) / Len ).

SSx18 is SumXsq18 - ( (SumX18 * SumX18) / Len ).

PearR18 is SP18/SQRT(SSx18*SSy).

PearRsq18 is ( PearR18 * PearR18 ).


ChgOCL19 is TickerUDF(L19, ChgOC).

SumXsq19 is Sum(ChgOCL19*ChgOCL19, Len).

SumX19 is Sum(ChgOCL19, Len).

SumXY19 is Sum(ChgOC*ChgOCL19, Len).

SP19 is SumXY19 - ( (SumX19 * SumY) / Len ).

SSx19 is SumXsq19 - ( (SumX19 * SumX19) / Len ).

PearR19 is SP19/SQRT(SSx19*SSy).
```

```
PearRsq19 is ( PearR19 * PearR19 ).


ChgOCL20 is TickerUDF(L20, ChgOC).

SumXsq20 is Sum(ChgOCL20*ChgOCL20, Len).

SumX20 is Sum(ChgOCL20, Len).

SumXY20 is Sum(ChgOC*ChgOCL20, Len).

SP20 is SumXY20 - ( (SumX20 * SumY) / Len ).

SSx20 is SumXsq20 - ( (SumX20 * SumX20) / Len ).

PearR20 is SP20/SQRT(SSx20*SSy).

PearRsq20 is ( PearR20 * PearR20 ).


!REPORTS TO DISPLAY VALUES"

PearsonR if 1=1.

PearsonRsq if 1=1.
```

The code provided will compute the Pearson correlation or "*r*" between two pairs of stocks for up to 20 stocks at a time. The EDS reports can be exported to Excel for further analysis. I used the AIQ Matchmaker module, which uses *Spearman rank* correlation to find highly correlated stocks from within the NASDAQ 100. I then created a list of these 20 symbols and set the EDS file to run only on this list. I then typed in the list in the same alphabetic order as they are in the list. I then set the report date to 1/6/2014, which is about one year prior to when I was writing this. The correlations are set to look back 250 days based on the input "len." Once we have decided how we will trade the basket created by the correlation analysis, we would forward-test using data not used to correlate the stocks, or from 1/7/2014 to 1/8/2015.



Symbol	PearR1	PearR2	PearR3	PearR4	PearR5	PearR6	PearR7	PearR8
ADBE	1.00	0.43	0.39	0.33	0.28	0.38	0.45	0.34
ADI	0.43	1.00	0.43	0.38	0.59	0.59	0.46	0.31
ADP	0.39	0.43	1.00	0.29	0.36	0.37	0.39	0.33
AKAM	0.33	0.38	0.29	1.00	0.27	0.31	0.29	0.26
ALTR	0.28	0.59	0.36	0.27	1.00	0.56	0.27	0.24
AMAT	0.38	0.59	0.37	0.31	0.56	1.00	0.30	0.31
FOX	0.45	0.46	0.39	0.29	0.27	0.30	1.00	0.29
INTU	0.34	0.31	0.33	0.26	0.24	0.31	0.29	1.00
LLTC	0.38	0.66	0.37	0.37	0.51	0.49	0.35	0.29
LMCA	0.30	0.35	0.34	0.28	0.28	0.33	0.42	0.22
LRCX	0.36	0.54	0.40	0.32	0.51	0.68	0.28	0.23
LVNTA	0.37	0.29	0.32	0.28	0.24	0.25	0.33	0.21
NTAP	0.14	0.28	0.21	0.20	0.27	0.12	0.15	0.14
NVDA	0.30	0.48	0.33	0.40	0.37	0.43	0.29	0.25
NXPI	0.30	0.56	0.25	0.28	0.51	0.41	0.32	0.24
PAYX	0.34	0.35	0.57	0.29	0.19	0.38	0.28	0.20
SIRI	0.27	0.33	0.27	0.38	0.22	0.23	0.35	0.19
TRIP	0.32	0.29	0.33	0.30	0.23	0.21	0.33	0.23
TXN	0.36	0.67	0.45	0.27	0.53	0.58	0.37	0.30
VIAB	0.36	0.44	0.46	0.34	0.25	0.40	0.60	0.28

FIGURE 5: AIQ, CORRELATION REPORT. This shows a portion of the correlation matrix report that is created from the AIQ EDS code for 20 of the NASDAQ 100 stocks in my sample test.

Figure 5 show s a portion of the resulting report. The cells w ith a 1.00 value are the result of correlating the stock w ith itself. Since the column headings use a variable name, the cells w ith the 1.00 value identify the stock in that column. This matrix can be used to create the correlated baskets described by the author. I also created a report for Pearson's *r* squared, but this is not used by Cline in his article.

—Richard Denning
info@TradersEdgeSystems.com
for AIQ Systems

BACK TO LIST

TRADERSSTUDIO: APRIL 2015

The TradersStudio code for Dave Cline's article in this issue, "Basket Trading Using A Directed Acyclic Graph," can be found at:

- www.TradersEdgeSystems.com/traderstips.htm
- www.TradersStudio.com → Traders Resources

The following code file is provided in the download:

> System: BASKET—A long-only system that uses daily data and buys all stocks that are set up in the session data list. There are no exits, as this is used solely for getting the complete set of correlation values for each pair.

The purpose of the very simple code I am providing is to get the "Eq Correlation" report to run on the entire period based on the start/end dates input in the session backtest for the entire list of stocks in the session. Note that the correlation report is based on a proprietary correlation formula that is similar to Pearson's *r*.

The TradersStudio code is as follows:

```
'BASKET TRADING USING A DIRECTED ACYCLIC GRAPH

'Author: Dave Cline, TASC April 2015

'Coded by: Richard Denning 2/10/2015

'www.TradersEdgeSystems.com



sub BASKET()

If 1=1 Then Buy("LE",1,0,Market,Day)

End Sub
```

Figure 6 shows a portion of the EQ Correlation report that I ran on the NASDAQ 100 stocks for the period from 1/6/2013 to 1/6/2014. The cells with a "1" value are the result of correlating the stock with itself. Since the column headings use a variable name, the cells with the "1" value identify the stock in that column. This matrix can be used to create the correlated baskets described by the author. The report can be saved to an Excel file for further analysis of the pairs.

Correlation Report for Session				
BASKET_ses Consolidated 1/6/2013 to 1/6/2014. System is BASKET()				
Period	Week			
*	1	2	3	4
AAPL.CSV - 1	1	0.0327	0.1145	0.0474
ADBE.CSV - 2	0.0327	1	0.3316	0.1607
ADI.CSV - 3	0.1145	0.3316	1	0.4735
ADP.CSV - 4	0.0474	0.1607	0.4735	1
ADSK.CSV - 5	0.2254	0.2961	0.4090	0.2373
AKAM.CSV - 6	0.1819	0.1382	0.4567	0.2371
ALTR.CSV - 7	0.1153	0.2462	0.5851	0.3480
ALXN.CSV - 8	0.1128	0.0824	0.2822	0.2582
AMAT.CSV - 9	0.1656	0.1953	0.4164	0.1998
AMGN.CSV - 10	-0.3033	0.0495	0.0242	0.2448

FIGURE 6: TRADERSSTUDIO, CORRELATION REPORT. This shows a portion of the EQ Correlation report that I ran on the NASDAQ 100 stocks for the period from 1/6/2013 to 1/6/2014.

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

UPDATA: APRIL 2015

Our Traders' Tip for this month is a system found in the Updata library, named *breadth correlation system*.

This system measures the average correlation of S&P 500 Index constituents to the VIX Index future. If this correlation falls below a certain level, then Donchian channels are used to give signals for market entry, assuming that the cyclical correlation value will eventually rise, and that periods of high correlation are also trending phases for the market. Exits are on a fixed number of days, proportional in number to the Donchian period.

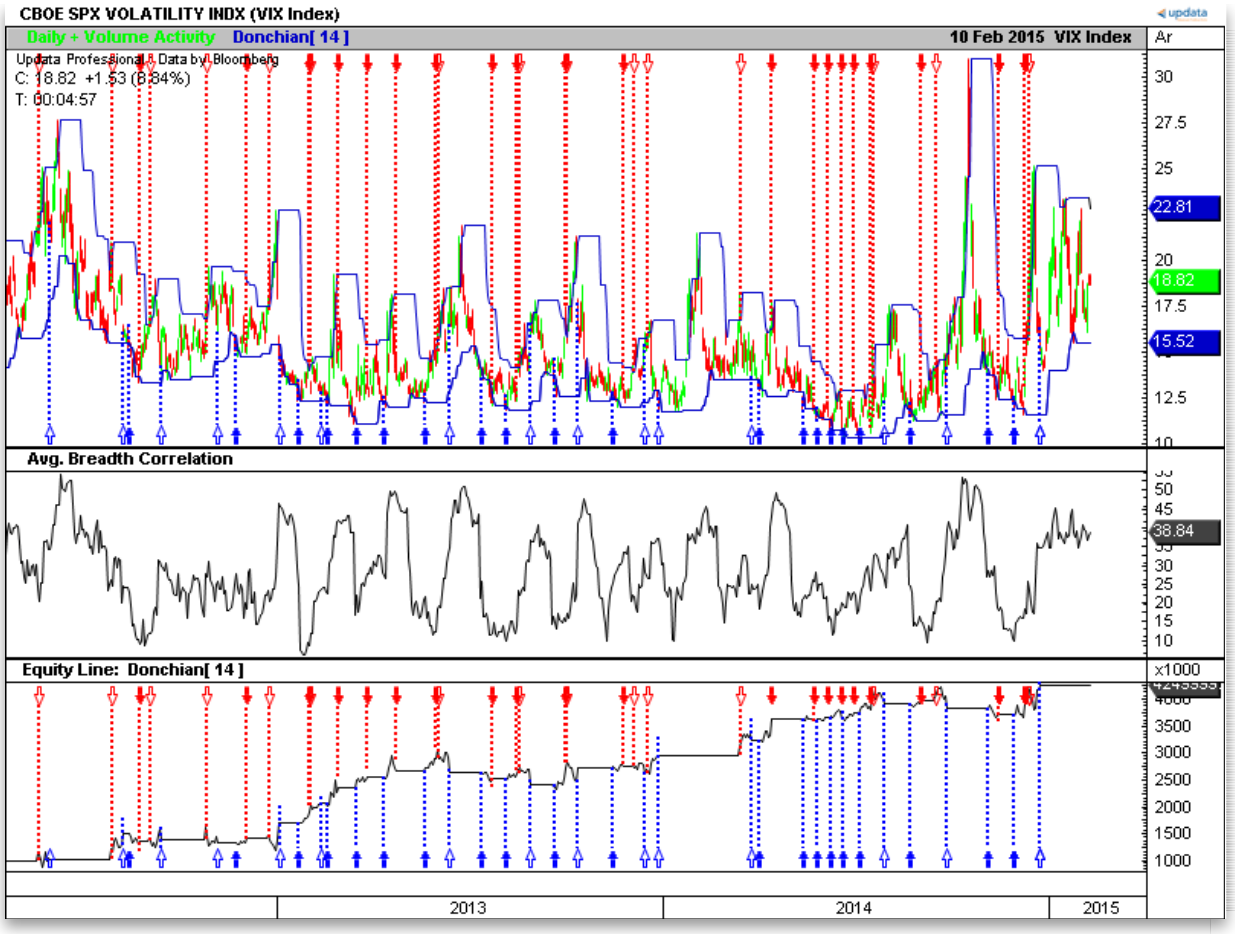


FIGURE 7: UPDATA, BREADTH CORRELATION. This chart shows the breadth correlation system as applied to the rolling VIX future, of daily resolution.

The Updata code for this article can be found in the Updata library and may be dow nloaded by clicking the *custom* menu and *system library*. Those w ho cannot access the library due to a firew all may paste the code show n here into the Updata custom editor and save it.

```
PARAMETER "WatchList" &MyList=SELECT
PARAMETER "Period" #PERIOD=14
PARAMETER "RsQ Thresh." @THRESH=30
DISPLAYSTYLE 3LINES
INDICATORTYPE TOOL
INDICATORTYPE3 CHART
COLOUR RGB(0,0,200)
COLOUR2 RGB(0,0,200)
NAME "Donchian[" #PERIOD "]" ""
NAME3 "Avg. Breadth Correlation" ""
#TICKERNUMBER=0
#COUNT=0
@CORRELSUM=0
@PeriodUpper=0
@PeriodLower=0
FOR #CURDATE=#PERIOD TO #LASTDATE
   'VARIABLES
   #COUNT=0
   @CORRELSUM=0
```

```
   @PeriodUpper=PHIGH(HIGH(1),#PERIOD,1)
   @PeriodLower=PLOW(LOW(1),#PERIOD,1)
   'AVG THE CORRELATION STATISTIC ACROSS THE WATCHLIST
   FOR #TICKERNUMBER=0 to LENGTH(&MyList)-1
       @CORRELSUM=@CORRELSUM+COEFFDET(CLOSE,CLOSE(&
MyList,#TICKERNUMBER),#PERIOD)/(LENGTH(&MyList)-1)
   NEXT
   'EXIT AFTER #PERIOD/2 BARS HELD
   IF ORDEROPENFOR>INT(#PERIOD/2)
      COVER CLOSE
      SELL CLOSE
   ENDIF
   'SHORT ENTRY
   IF @CORRELSUM<@THRESH AND CLOSE<@PeriodLower
      BUY CLOSE
   ENDIF
   'LONG ENTRY
   IF @CORRELSUM<@THRESH AND CLOSE>@PeriodUpper
      SHORT CLOSE
   ENDIF
   'PLOT LINES
   @PLOT3=@CORRELSUM
   @PLOT=@PeriodUpper
   @PLOT2=@PeriodLower
NEXT
```

—Updata support team
support@updata.co.uk, www.updata.co.uk

BACK TO LIST

MICROSOFT EXCEL: APRIL 2015

The article "Basket Trading Using A Directed Acyclic Graph" by Dave Cline in this issue is the proverbial duck on a pond: At first glance, it's an interesting introduction to basket trading. But under the surface, a closer reading show s there is a lot of w ork that must go into preparing our trading basket before w e can start trading.

Cline's article describes three rather sizeable pieces:

1. Build a raw , pairw ise correlation database from a user-supplied list of symbols.

a. Set a time w indow for sampling symbol correlations. This should be built on recent history for a live trading system. How ever, for backtesting, *use period chosen* should predate the intended backtest period.
b. Set the number of bars to be used for correlation calculations.
c. Set a correlation value threshold to filter out the w orst-performing correlation pairs.
d. Perhaps further limit the list to the *N* pairs w ith the highest correlation values.

2. Use *directed acyclic graph* (DAG) processing to trim the *correlations* list built above into a sizeable list of symbols w ith strong mutual intercorrelations.

Notice that step 1 may give us "islands" of symbols w ith strong intercorrelations w ithin the island. I call them

islands because symbols in these groupings, while strongly intercorrelated among members of the containing island, will have no strong intercorrelations with symbols in any of the other islands. This is because any interconnections that may have existed between the separate groupings did not survive the correlation threshold filtering in step 1. In short, the islands do not correlate well with each other. They move in different ways.

Any one of these islands could be the basis of a basket trading group.

DAG critical path processing is intended to choose the largest possible set of the most strongly intercorrelated symbols; presumably, this set will come out of the island group with the strongest internal correlations.

Cline takes this one step further using the custom method GetLimitedNetworkedList as the last step of the DAG process to apply some additional filtering refinements.

3. Feed the resulting list of symbols to a basket trading system of your choice.

Any one of these three pieces represents a significant spreadsheet on its own. And none of them lends itself to being done purely, or even mostly, as spreadsheet cell formulas. Thus, the spreadsheet for this Traders' Tip uses lots of VBA macro code.

The spreadsheet presented here tackles the first piece: *building the correlation pairs database from a user-specified list of symbols*.

My Figures 8–11 show steps of the process. My example uses a list of the symbols that constitute the S&P 500 index.

1. On the Internet, look up the constituent symbols list for the S&P 500.

a. Copy to a work area for manual cleanup efforts. In this case, a worksheet tab named "S&P 500 constituent symbols" serves as my work area.
b. Manually remove any embedded links that came with the copied list.
c. Manually remove extraneous stuff that also came with the copied list.
d. Adjust the symbol formats. A couple of the symbols contained periods, such as BRK.B and BF.B. Replace the periods with hyphens, which is the Yahoo Finance standard.

(See Figure 8.)

Stocks and Indexes for which I want pair-wise correlation values		Correlation Period	rough in:	01/01/2011	to	01/01/2012	<= hard end date				**Symbol retrieval stats**				
Only need list of valid symbols in column A starting in row 10.				250	Number of bars I want to use as the correlation window.						502	Candidate Symbols			
All of the other fields will be filled in as the data for each symbol is retrieved.				08/30/2010	Earliest date to retrieve (with 100 days of padding).					100.00%	502	Completed	00:00:02.125	Average Elapsed / symbol	
					Data Retrieval Start:		2/11/2015 0:27:39	Elapsed			0	Remaining	00:00:00.000	Est. Time Remaining	
483	Symbols with at least 250 bars.		Generate Pairwise Correlations		Data Retrieval End:		2/11/2015 0:45:26	00:17:47			**Correlation Stats**		2/11/2015 0:45:26	ETA	
					Correlation End:		2/11/2015 0:59:32	00:14:06			116,403	Possible pairings when using 483 symbols.			
					Total Elapsed:		00:31:53			100.00%	116,403	Completed	00:00:00.007	Average Elapsed per Pairing	
Symbol To Correlate	**Description**	**Last Trade as of most recent check**								**Bars Available**	0	Remaining	00:00:00.000	Est time remaining	
		Date	**Time**	**Open**	**High**	**Low**	**Close**	**Volume**	**Change**				2/11/2015 0:59:32	ETA	
MMM	3M Company Common (MMM) -NYSE (Daily)	02/10/2015	4:01 PM	165.40	166.14	164.60	165.82	1,885,097	0.980	339					
ABT	Abbott Laboratori (ABT) -NYSE (Daily)	02/10/2015	4:00 PM	44.68	45.01	44.40	44.92	5,210,459	0.400	339					
ABBV	AbbVie Inc. Commo (ABBV) -NYSE (Daily)	02/10/2015	4:00 PM	55.94	57.01	55.60	56.89	18,916,988	1.410	1					
ACE	Ace Limited Commo (ACE) -NYSE (Daily)	02/10/2015	4:01 PM	112.75	113.00	111.63	111.75	1,213,513	-0.220	339					
ACN	Accenture plc. Cl (ACN) -NYSE (Daily)	02/10/2015	4:01 PM	87.74	87.91	86.68	87.52	2,573,782	0.160	339					
ACT	Actavis plc Ordin (ACT) -NYSE (Daily)	02/10/2015	4:03 PM	274.79	279.76	272.78	279.14	2,407,406	6.830	339					
ADBE	Adobe Systems Inc (ADBE) -NasdaqNM (Daily)	02/10/2015	4:00 PM	72.94	73.19	72.11	72.76	1,696,213	0.430	339					
ADT	ADT Corporation C (ADT) -NYSE (Daily)	02/10/2015	4:00 PM	37.43	37.55	37.08	37.30	1,674,953	0.180	1					
AES	The AES Corporati (AES) -NYSE (Daily)	02/10/2015	4:00 PM	11.74	12.20	11.53	12.16	11,225,954	0.310	339					
AET	Aetna Inc. Common (AET) -NYSE (Daily)	02/10/2015	4:02 PM	94.15	95.95	94.04	95.75	1,900,646	2.010	339					
AFL	AFLAC Incorporate (AFL) -NYSE (Daily)	02/10/2015	4:02 PM	61.19	61.58	60.89	61.45	1,677,607	0.310	339					
AMG	Affiliated Manage (AMG) -NYSE (Daily)	02/10/2015	4:00 PM	217.00	217.88	212.93	213.25	691,636	-1.500	339					
A	Agilent Technolog (A) -NYSE (Daily)	02/10/2015	4:03 PM	39.31	39.76	38.78	39.67	1,408,747	0.630	339					
GAS	AGL Resources, In (GAS) -NYSE (Daily)	02/10/2015	4:00 PM	54.06	54.62	53.83	54.50	1,083,386	0.400	339					
APD	Air Products and (APD) -NYSE (Daily)	02/10/2015	4:06 PM	150.02	150.86	148.65	150.29	915,605	0.520	339					
ARG	Airgas, Inc. Comm (ARG) -NYSE (Daily)	02/10/2015	4:03 PM	114.49	114.99	113.33	114.54	284,109	0.560	339					
AKAM	Akamai Technologi (AKAM) -NasdaqNM (Daily)	02/10/2015	4:00 PM	61.30	62.37	60.30	62.21	3,562,368	1.250	339					
AA	Alcoa Inc. Common (AA) -NYSE (Daily)	02/10/2015	4:00 PM	15.72	15.92	15.55	15.91	25,138,096	0.260	339					
ALXN	Alexion Pharmaceu (ALXN) -NasdaqNM (Daily)	02/10/2015	4:00 PM	171.15	177.37	171.11	176.39	1,334,985	4.970	339					
ATI	Allegheny Technol (ATI) -NYSE (Daily)	02/10/2015	4:03 PM	32.01	32.25	31.13	31.83	819,043	-0.390	339					
ALLE	Allegion plc Ordi (ALLE) -NYSE (Daily)	02/10/2015	4:00 PM	57.22	57.35	56.26	56.88	527,416	-0.030	1					
AGN	Allergan, Inc. Co (AGN) -NYSE (Daily)	02/10/2015	4:00 PM	223.81	226.76	223.62	226.54	3,210,038	3.430	339					
ADS	Alliance Data Sys (ADS) -NYSE (Daily)	02/10/2015	4:00 PM	276.45	280.22	275.32	279.44	606,406	6.320	339					
ALL	Allstate Corporat (ALL) -NYSE (Daily)	02/10/2015	4:00 PM	70.55	70.59	69.65	70.05	3,063,724	0.020	339					
ALTR	Altera Corporatio (ALTR) -NasdaqNM (Daily)	02/10/2015	4:00 PM	33.98	34.37	33.77	34.34	2,001,865	0.580	339					
MO	Altria Group, Inc (MO) -NYSE (Daily)	02/10/2015	4:00 PM	53.50	53.84	53.37	53.77	4,500,821	0.610	339					
AMZN	Amazon.com, Inc. (AMZN) -NasdaqNM (Daily)	02/10/2015	4:00 PM	370.89	374.30	368.51	373.00	2,287,232	2.435	339					
AEE	Ameren Corporatio (AEE) -NYSE (Daily)	02/10/2015	4:05 PM	42.27	43.39	42.19	43.34	2,071,787	1.140	339					
AEP	American Electric (AEP) -NYSE (Daily)	02/10/2015	4:01 PM	58.26	59.43	58.26	59.31	3,403,724	1.070	339					
AXP	American Express (AXP) -NYSE (Daily)	02/10/2015	4:01 PM	85.69	85.97	85.12	85.84	3,589,926	0.860	339					
AIG	American Internat (AIG) -NYSE (Daily)	02/10/2015	4:02 PM	52.53	52.77	51.96	52.41	5,580,829	0.270	339					
AMT	American Tower Co (AMT) -NYSE (Daily)	02/10/2015	4:00 PM	97.17	97.45	95.13	96.37	2,965,564	-0.330	339					

FIGURE 8: EXCEL, CORRELATION CANDIDATES. This shows the correlation candidates tab after a complete symbol retrieval and pair generation.

2. The *CorrelationCandidates* tab drives the process of dow nloading data for each of the candidate symbols. When the symbol dow nload phase is complete, the generation of the raw correlation pairs list automatically begins on the *RawCorrelations* tab.

User steps:

a. On the *CorrelationCandidates* tab, manually ClearContents of all row s below the header row (that is, row 8 dow n).
b. Manually copy in the list of symbols you cleaned up in step 1 starting in cell A8.
c. Set the correlation bar count, w hich is the number of bars required from each symbol of a pair for the correlation calculation.
d. Set begin and end dates for the correlation period of interest. The end date is used directly along w ith the correlation bar count plus a 100-bar padding to calculate the earliest date to retrieve. Yahoo history retrievals w ill give us all available bars for the period marked out by the earliest and the end dates.
e. Click the “Generate pairw ise correlations” button and go get lunch. This can take a w hile.

On my computer, late at night w hen it might be expected that the Yahoo servers and the Internet w ould perhaps be under less strain, the symbol retrieval phase (Figure 9) took almost 18 minutes for the 502 symbols in my list.

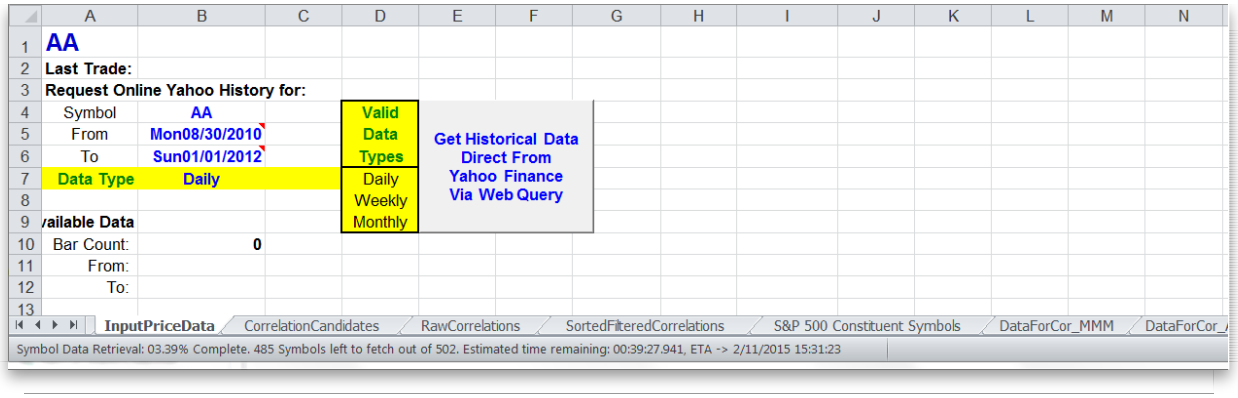


FIGURE 9: EXCEL, SYMBOL RETRIEVAL. Symbol retrieval is in flight. The status bar shows the phase progress and estimated time to completion.

The subsequent Correlation Pair Generation Phase (Figure 10) took an additional 14 minutes. Total elapsed time was about 31 minutes. Your mileage may vary.

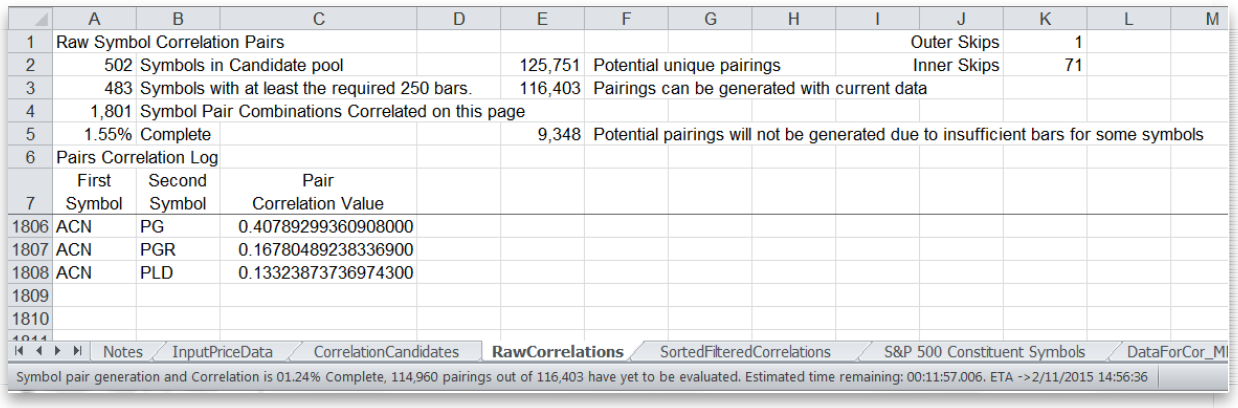


FIGURE 10: EXCEL, CORRELATION PAIR GENERATION. Here is the correlation pair generation phase with progress and estimated time to completion in the status bar.

The macros that manage the two phases of this work place progress messages on the bottom status bar. This computation uses the average time for the items already processed in the current phase to project a time remaining and an ETA for completion of the phase.

3. For the last filtering step, go to the *SortedFilteredCorrelation* tab. This is very quick, so playing with what-if scenarios at this stage is not at all painful.

Sort and Filter by Threshold and Max Entries

0.72 Absolute Correlation Threshold. Can specify out to 15 digits if you so choose

500 Max entries. A **non-zero** value is an absolute cut off for the number of result entries

27,131 Raw Entries Satisfied the Correlation Threshold

Get Top (Strongest) Correlations

Get Bottom (Weakest) Correlations

500 Pairs in this Filtered Correlation List

First Symbol	Second Symbol	Pair Correlation Value
ED	SO	0.99125617590455700
BAC	GS	0.98945553246631100
HIG	MET	0.98822233445254100
C	MS	0.98788082217714800
SEE	STI	0.98767122871828100
BK	MS	0.98766585676833000
GS	MS	0.98744400149188500

FIGURE11: EXCEL, READY FOR DAG PROCESSING. Here, the sorted and filtered correlation pairs are ready for DAG processing.

a. Set the two filter values to your taste (Figure 11).
   i. *Threshold* is a value between 1 and -1.
   ii. A *max entries* value greater than zero says that if the threshold filter allows a sufficiently large result set, trim the result to the top MaxEntries pairs. A zero value says give me anything the threshold value passes.
b. Normal, highest correlation results are obtained by clicking on the *get top correlations* button. Here is what is happening in the spreadsheet:
   i. Entries are sorted by correlation value in descending order.
   ii. Entries with a correlation value less than the threshold are removed from the bottom.
   iii. If Max Entries is not zero, then any entries more than Max Entries down the list are removed.
c. If you wish, the lowest correlation results may be obtained by clicking on the "Get Bottom Correlations" button.
   i. Entries are sorted by correlation value in ascending order, weakest at the top.
   ii. Flip the sign on the threshold (-threshold).
   iii. Entries with a correlation value greater than the *-threshold* value are removed.
   iv. If Max Entries is not zero, then any entries more than Max Entries down the list are removed.

4. Finally, export this refined list to your DAG processor of choice to complete selection of a basket of symbols.

After a full S&P 500 run, the saved spreadsheet takes around 17 megabytes. To reduce this spreadsheet to a manageable download size, I have stripped out retrieved symbol data. The download should be something less than 800 kb stored on your drive.

For demonstration purposes, you can cut back on the number of symbols you place on the CorrelationCandidates tab. Ten to 20 symbols should make a reasonable demo run. And a sample run of that size will not eat as much wall time while it runs, or hard drive space when you store it.

The spreadsheet file for this Traders' Tip can be downloaded here. To successfully download it, follow these steps:

- Right-click on the Excel file link, then
- Select "save as" (or "save target as") to place a copy of the spreadsheet file on your hard drive.

—Ron McAllister
Excel and VBA programmer

rpmac_xltt@sprynet.com

BACK TO LIST

May 2015

For this month's Traders' Tips, the focus is Giorgos Siligardos' article in this issue, "Filtering Price Movement." Here, we present the May 2015 Traders' Tips code with possible implementations in various software.

Code for MetaStock is already provided by Siligardos in his article, which S&C subscribers will find in the Subscriber Area of our website here.

The Traders' Tips section is provided to help the reader implement a selected technique from an article in this issue or another recent issue. The entries here are contributed by software developers or programmers for software that is capable of customization.

TRADESTATION: MAY 2015

In "Filtering Price Movement" in this issue, author Giorgos Siligardos describes a process to analyze historical market data. He begins by describing a concept that he refers to as *perceptually important points* (PIPs), which he uses to identify price extremes. From these, he creates an indicator he calls zzTOP that identifies these points and then connects them using trendlines.

For convenience, we're providing the code for a TradeStation indicator based on the author's description.

```
using elsystem;
using elsystem.collections ;
using elsystem.drawingobjects ;
using elsystem.drawing ;

inputs:
int Iterations( 3 );

variables:
        Vector BarData ( NULL ), // Bar DTP
        Vector Segments ( NULL ); // of Rounds of Pairs

method void CreateTL ( DTPoint Begin_DTP,
        DTPoint End_DTP )
        variables: TrendLine New_TL ;
        begin
```

```
        New_TL = TrendLine.Create( Begin_DTP, End_DTP ) ;
        New_TL.ExtLeft = false ;
        New_TL.ExtRight = false ;
        New_TL.Persist = true ;
        New_TL.Color = Color.Aquamarine ;
        DrawingObjects.Add( New_TL ) ;
        end ;

method void LoadBarDataVector ()
        begin
        if BarData <> NULL then
                BarData.Push_Back(
                        DTPoint.Create( BarDateTime, Close ) astype DTPoint ) ;
        end ;

method DTPoint BisectPairs ( Vector DTPPair )
        variables: DTPoint BegDTP, DTPoint EndDTP,
                        DTPoint MidDTP,
                        int BarDataStartIndex, int BarDataEndIndex,
                        int BarDataCnt,
                        int HighDiffIndex, double HighDiff,
                        int Count, double TL_Val  ;
        begin
        HighDiffIndex = 0 ;
        HighDiff = -999999 ;

        BegDTP = DTPPair.Items[0] astype DTPoint ;
        EndDTP = DTPPair.Items[1] astype DTPoint ;

        for Count = 0 to BarData.Count - 1
                begin
                if (BarData.Items[Count] astype DTPoint).DateTimeOfBar =
                        BegDTP.DateTimeOfBar then
                        BarDataStartIndex = Count ;
                if (BarData.Items[Count] astype DTPoint).DateTimeOfBar =
                        EndDTP.DateTimeOfBar then
                        BarDataEndIndex = Count ;
                end ;

        BarDataCnt = BarDataEndIndex - BarDataStartIndex ;
        if BarDataCnt = 0
                then BarDataCnt +=1 ;
        for Count = BarDataStartIndex to BarDataEndIndex
                begin
                // Find TL Value
                TL_Val = (BarData.Items[BarDataStartIndex]
                        astype DTPoint).Price astype double +
                        (( BarData.Items[BarDataEndIndex] astype DTPoint).Price
astype double -
                        (BarData.Items[BarDataStartIndex] astype DTPoint).Price
astype double )
                        * ( ( Count ) / BarDataCnt ) ;
                if AbsValue( TL_Val - (BarData.Items[Count] astype DTPoint).
                        Price astype double ) > HighDiff then
                        begin
                        HighDiff = AbsValue( TL_Val -
                                (BarData.Items[Count] astype DTPoint).Price astype
double ) ;
                        HighDiffIndex = Count ;
                        end ;
```

```
                end ;
        MidDTP = BarData.Items[HighDiffIndex] astype DTPoint ;
        return MidDTP ;
        end;

method void findDiffBisect( )
        variables: Vector RecursRound, Vector NewRound,
                                Vector NewPairR, Vector NewPairL,
                                int Count, bool OKToUse,
                                DTPoint BegDTP, DTPoint EndDTP,  DTPoint MidDTP ;
        begin
        NewRound = new Vector ;
        RecursRound = new Vector ;
        OKToUse = false ;
        RecursRound = segments.Items[Segments.Count -1] astype Vector ;
        for Count = 0 to RecursRound.Count -1
                begin
                        begin
                        MidDTP = BisectPairs( RecursRound.Items[Count] astype Vector
) ;
                        NewPairL = new Vector ;
                        NewPairL.Push_Back( (RecursRound.Items[Count]
                                astype Vector).Items[0] astype DTPoint ) ;
                        NewPairL.Push_Back(  MidDTP astype DTPoint ) ;
                        NewRound.Push_Back( NewPairL astype Vector ) ;
                        NewPairR = new Vector ;
                        NewPairR.Push_Back(  MidDTP astype DTPoint ) ;
                        NewPairR.Push_Back( (RecursRound.Items[Count]
                                astype Vector).Items[1] astype DTPoint ) ;
                        NewRound.Push_Back( NewPairR astype Vector ) ;
                        end ;
                end ;
        Segments.push_back( NewRound ) ;
        end ;

method void SeedSegmentsVector()
        variables: Vector RecursRound, Vector Pair ;
        begin
        Pair = new Vector ;
        RecursRound = new Vector ;
        Pair.Push_Back( BarData.Items[0] astype DTPoint ) ;
        Pair.Push_Back( BarData.Items[BarData.Count -1] astype DTPoint ) ;
        RecursRound.Push_Back( Pair astype Vector ) ;
        Segments.Push_Back( RecursRound astype Vector ) ;
        end ;

method void DrawAllTLs( )
        variables: Vector RecursRound, Vector Pair, int Count ;
        begin
        Pair = new Vector ;
        RecursRound = new Vector ;
        RecursRound = Segments.Items[Segments.Count -1] astype vector ;
        Print( "TLStart" ) ;
        for Count = 0 to RecursRound.Count -1
                begin
                Pair = RecursRound.Items[Count] astype Vector ;
                CreateTL( Pair.Items[0] astype DTPoint, Pair.Items[1] astype DTPoint
);
                end ;
        end ;
```

```
once
        begin
        ClearPrintLog ;
        BarData = new Vector ;
        Segments = new Vector ;
        end ;

if BarStatus( DataNum + 1 ) = 2 then
        LoadBarDataVector () ;

once ( LastBarOnChartEx )
        begin
        SeedSegmentsVector( ) ;
        For Value1 = 1 to Iterations
                begin
                findDiffBisect(  ) ;
                end ;
        DrawAllTLs( ) ;
        end ;
```

To dow nload the EasyLanguage code, please visit our TradeStation and EasyLanguage support forum. The code can be found here: http://w w w .tradestation.com/TASC-2015. The ELD filename is "_TASC_ModifiedTrueRange.ELD." For more information about EasyLanguage in general, please see http://w w w .tradestation.com/EL-FAQ.

A sample chart is show n in Figure 1.

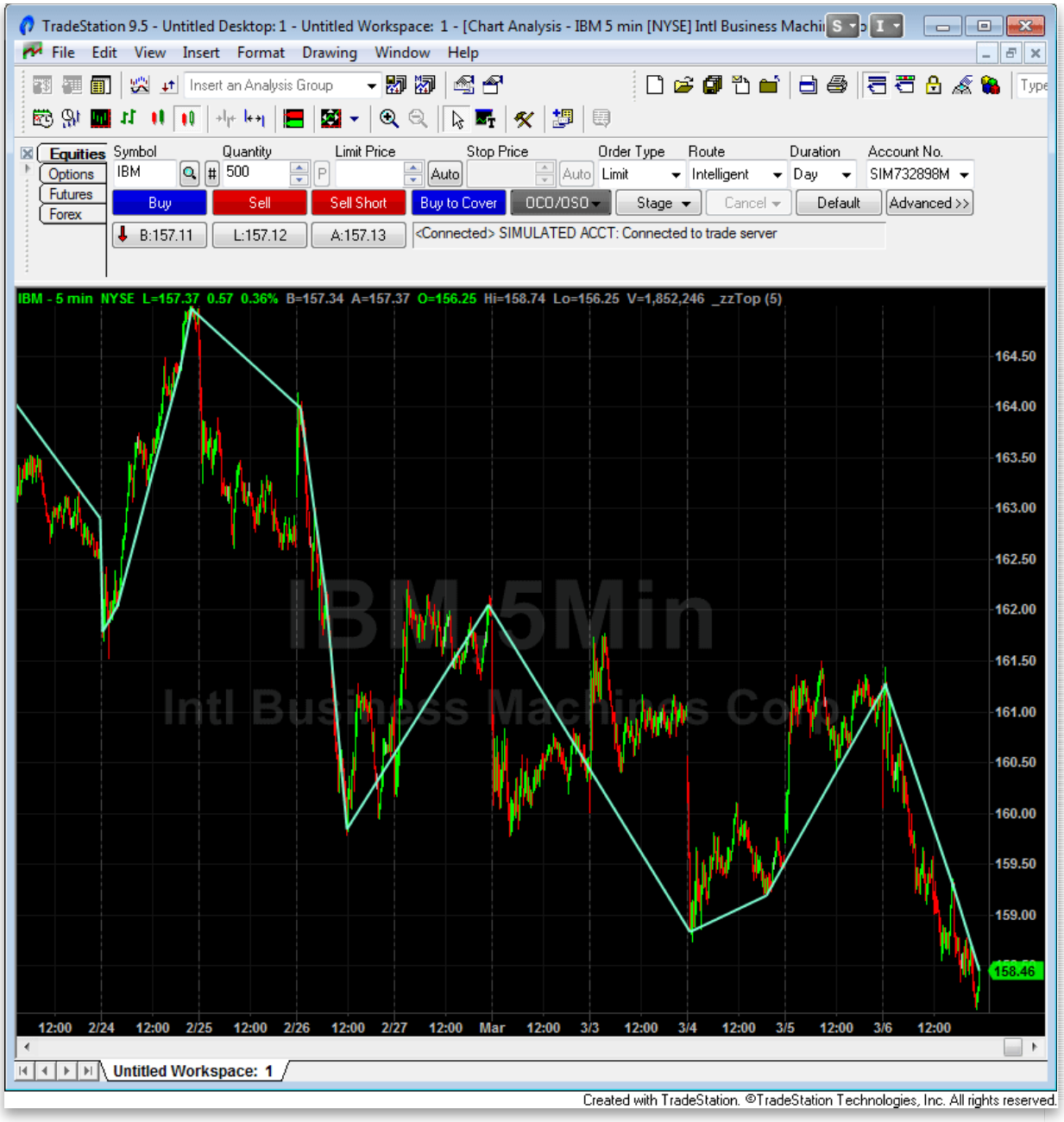


FIGURE 1: TRADESTATION. Here, the zzTop indicator is applied to an intraday chart of IBM.

This article is for informational purposes. No type of trading or investment recommendation, advice, or strategy is being made, given, or in any manner provided by TradeStation Securities or its affiliates.

—Doug McCrary
TradeStation Securities, Inc.
www.TradeStation.com

BACK TO LIST

eSIGNAL: MAY 2015

For this month's Traders' Tip, w e've provided the formulas zzTOP.efs and zzTOPauto.efs based on the formulas described in Giorgos Siligardos' article in this issue, "Filtering Price Movement."

The studies contain formula parameters to set the desired period and price, w hich may be configured through the *edit chart* w indow (right-click on the chart and select "edit chart") to set the desired period and price. A sample chart is show n in Figure 2.

FIGURE 2: eSIGNAL. Here is an example of the study implemented on a chart of Boston Scientific Corp. (BSX).

To discuss these studies or dow nload a complete copy of the formulas' code, please visit the EFS Library Discussion Board forum under the *forums* link from the support menu at w w w .esignal.com or visit our EFS Know ledgeBase at http://w w w .esignal.com/support/kb/efs/. The eSignal formula scripts (EFS) are also available for copying & pasting below .

---

zzTOP.efs:

```
/*********************************
Provided By:
    Interactive Data Corporation (Copyright B© 2015)
    All rights reserved. This sample eSignal Formula Script (EFS)
    is for educational purposes only. Interactive Data Corporation
    reserves the right to modify and overwrite this EFS file with
    each new release.
```

```
Description:
    Filtering Price Movement by Giorgos E. Siligardos

Formula Parameters:                     Default:
Indicator                               Close
LegsNo                                  20
Scale                                   A

Version:            1.00  03/11/2015

Notes:
    The related article is copyrighted material. If you are not a subscriber
    of Stocks & Commodities, please visit www.traders.com.

**********************************/

var fpArray = new Array();

function preMain(){

    setStudyTitle("zzTOP");
    setPriceStudy(true);
    setComputeOnClose(true);

    var x = 0;

    fpArray[x] = new FunctionParameter("fpIndicator", FunctionParameter.STRING);
    with(fpArray[x++]){
        setName("Indicator");
        addOption("Close");
        addOption("Open");
        addOption("High");
        addOption("Low");
        setDefault("Close");
    }

    fpArray[x] = new FunctionParameter("fpLegsNo", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("LegsNo");
        setLowerLimit(1);
        setDefault(20);
    }

    fpArray[x] = new FunctionParameter("fpScale", FunctionParameter.STRING);
    with(fpArray[x++]){
        setName("Scale");
        addOption("A");
        addOption("L");
        setDefault("A");
    }
}

var bInit = false;
var bVersion = null;

var xSourceBase = null;
var xSource = null;

function main(fpIndicator, fpLegsNo, fpScale){
```

```
    if (!bInit){

        switch (fpIndicator){
            case "Close":
                xSourceBase = close();
                break;
            case "Open":
                xSourceBase = open();
                break;
            case "High":
                xSourceBase = high();
                break;
            case "Low":
                xSourceBase = low();
                break;
            default: return;
        }

        if (fpScale == "L")
            xSource = efsInternal("calc_Log", xSourceBase)
        else
            xSource = xSourceBase;

        bInit = true;
    }

    if (getCurrentBarCount() == (getNumBars()-1)){

        var nCountOfBars = getCurrentBarCount();

        var nLowestInd = lowest(nCountOfBars, xSourceBase, 0);

        if (fpScale == "L" && nLowestInd != null && nLowestInd <= 0){
            drawTextPixel( 10, 50, "A semilogarithmic scale is not supported for the
negative values or 0.", Color.blue);
            return;
        }

        if (nLowestInd == null)
            return;

        var x1 = -(nCountOfBars - 1);
        var y1 = xSource.getValue(-(nCountOfBars - 1));
        var aFirstBar = [x1, y1];

        var x2 = 0;
        var y2 = xSource.getValue(0);
        var aLastBar = [x2, y2];

        if (y1 == null || y2 == null)
            return;

        var arrayPIPs = [];
        arrayPIPs.push(aFirstBar);
        arrayPIPs.push(aLastBar);

        while (arrayPIPs.length < fpLegsNo + 1){

            var arrayTemp = [];
```

```
            for (var i = 0; i < arrayPIPs.length - 1; i++){
                arrayTemp.push(calc_PIP(arrayPIPs[i], arrayPIPs[i+1], xSource))
            }

            var aMaxTempElement = arrayTemp.reduce(function(prevElemen, curElement){
                return prevElemen[0] >= curElement[0] ? prevElemen : curElement
            });

            var x = aMaxTempElement[1];
            var y = xSource.getValue(aMaxTempElement[1]);
            if (y == null)
                return;
            arrayPIPs.push([x, y]);

            arrayPIPs.sort(function (element1, element2){
                return element1[0] - element2[0];
            });
        }

        var tagID = 0;
        clearLines();

        for (var i = 0; i < arrayPIPs.length - 1; i++){
            x1 = arrayPIPs[i][0];
            y1 = arrayPIPs[i][1];

            x2 = arrayPIPs[i+1][0];
            y2 = arrayPIPs[i+1][1];

            if (fpScale == "L"){
                y2 = Math.exp(y2);
                y1 = Math.exp(y1);
            }

            drawLineRelative(x1, y1, x2, y2, PS_SOLID, 2, Color.red, tagID++);
        }
    }
}

function calc_Log(xSource){
    var nValue = xSource.getValue(0);

    if (nValue <= 0)
        return;

    return Math.log(nValue);
}

function calc_PIP(aFirstCoord, aSecondCoord, xSource){

    var nIndexOfFirst = aFirstCoord[0];
    var nIndexOfSecond = aSecondCoord[0];
    var nCountOfBars = Math.abs(nIndexOfFirst - nIndexOfSecond) + 1;

    var nPrevMaxDiff = 0;
    var nPrevMaxIndex  = 0;

    var aMaxValue = [];
```

```
    for (var i = 0; i < nCountOfBars; i++){

        var nIndicatorValue = xSource.getValue(nIndexOfFirst + i);
        if (nIndicatorValue == null)
            return;

        var nCrossPoint = (((i * (aSecondCoord[1] - aFirstCoord[1])) / (nCountOfBars
- 1)) + aFirstCoord[1]);
        var nDiff = Math.abs(nCrossPoint - nIndicatorValue);

        aMaxValue = [];
        aMaxValue[0] = Math.max(nDiff, nPrevMaxDiff);
        aMaxValue[1] = aMaxValue[0] == nDiff ? nIndexOfFirst + i : nPrevMaxIndex;

        nPrevMaxDiff = aMaxValue[0];
        nPrevMaxIndex  = aMaxValue[1];
    }
    return aMaxValue;
}

function verify(){

    var b = false;
    if (getBuildNumber() < 779){

        drawTextAbsolute(5, 35, "This study requires version 8.0 or later.",
            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,
            null, 13, "error");
        drawTextAbsolute(5, 20, "Click HERE to upgrade.@URL=http://www.esignal.com
/download/default.asp",
            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,
            null, 13, "upgrade");
        return b;
    }
    else
        b = true;

    return b;
}
```

---

zzTOPauto.efs:

```
/*********************************
Provided By:
    Interactive Data Corporation (Copyright B© 2015)
    All rights reserved. This sample eSignal Formula Script (EFS)
    is for educational purposes only. Interactive Data Corporation
    reserves the right to modify and overwrite this EFS file with
    each new release.

Description:
    Filtering Price Movement by Giorgos E. Siligardos

Formula Parameters:                     Default:
Indicator                               Close
Proximity                               20
```

```
Scale                                   A

Version:            1.00  03/11/2015

Notes:
    The related article is copyrighted material. If you are not a subscriber
    of Stocks & Commodities, please visit www.traders.com.

**********************************/

var fpArray = new Array();

function preMain(){

    setStudyTitle("zzTOPauto");
    setPriceStudy(true);
    setComputeOnClose(true);

    var x = 0;

    fpArray[x] = new FunctionParameter("fpIndicator", FunctionParameter.STRING);
    with(fpArray[x++]){
        setName("Indicator");
        addOption("Close");
        addOption("Open");
        addOption("High");
        addOption("Low");
        setDefault("Close");
    }

    fpArray[x] = new FunctionParameter("fpProximity", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("Proximity");
        setLowerLimit(0);
        setUpperLimit(100);
        setDefault(20);
    }

    fpArray[x] = new FunctionParameter("fpScale", FunctionParameter.STRING);
    with(fpArray[x++]){
        setName("Scale");
        addOption("A");
        addOption("L");
        setDefault("A");
    }
}

var bInit = false;
var bVersion = null;

var xSourceBase = null;
var xSource = null;

function main(fpIndicator, fpProximity, fpScale){

    if (!bInit){

        switch (fpIndicator){
            case "Close":
                xSourceBase = close();
```

```
                break;
            case "Open":
                xSourceBase = open();
                break;
            case "High":
                xSourceBase = high();
                break;
            case "Low":
                xSourceBase = low();
                break;
            default: return;
        }

        if (fpScale == "L")
            xSource = efsInternal("calc_Log", xSourceBase)
        else
            xSource = xSourceBase;

        bInit = true;
    }

    if (getCurrentBarCount() == (getNumBars()-1)){

        var nCountOfBars = getCurrentBarCount();

        var nLowestInd = lowest(nCountOfBars, xSourceBase, 0);

        if (fpScale == "L" && nLowestInd != null && nLowestInd <= 0){
            drawTextPixel( 10, 50, "A semilogarithmic scale is not supported for the
negative values or 0.", Color.blue);
            return;
        }

        if (nLowestInd == null)
            return;

        var nHighest = highest(nCountOfBars, xSource, 0);
        var nLowest = lowest(nCountOfBars, xSource, 0);

        if (nHighest == null || nLowest == null)
            return;

        var nRange = nHighest - nLowest;

        var x1 = -(nCountOfBars - 1);
        var y1 = xSource.getValue(-(nCountOfBars - 1));
        var aFirstBar = [x1, y1];

        var x2 = 0;
        var y2 = xSource.getValue(0);
        var aLastBar = [x2, y2];

        if (y1 == null || y2 == null)
            return;

        var arrayPIPs = [];
        arrayPIPs.push(aFirstBar);
        arrayPIPs.push(aLastBar);

        do {
```

```
            var arrayTemp = [];

            for (var i = 0; i < arrayPIPs.length - 1; i++){
                arrayTemp.push(calc_PIP(arrayPIPs[i], arrayPIPs[i+1], xSource));
            }

            var aMaxTempElement = arrayTemp.reduce(function(prevElemen, curElement){
                return prevElemen[0] >= curElement[0] ? prevElemen : curElement
            });

            if (aMaxTempElement[0] < (nRange * fpProximity / 100))
                break;

            var x = aMaxTempElement[1];
            var y = xSource.getValue(aMaxTempElement[1]);
            if (y == null)
                return;

            arrayPIPs.push([x, y]);

            arrayPIPs.sort(function (element1, element2){
                return element1[0] - element2[0];
            });

        } while (aMaxTempElement[0] >= (nRange * fpProximity / 100));

        var tagID = 0;
        clearLines();

        for (var i = 0; i < arrayPIPs.length - 1; i++){
            x1 = arrayPIPs[i][0];
            y1 = arrayPIPs[i][1];

            x2 = arrayPIPs[i+1][0];
            y2 = arrayPIPs[i+1][1];

            if (fpScale == "L"){
                y2 = Math.exp(y2);
                y1 = Math.exp(y1);
            }

            drawLineRelative(x1, y1, x2, y2, PS_SOLID, 2, Color.red, tagID++);
        }
    }
}

function calc_Log(xSource){
    var nValue = xSource.getValue(0);

    if (nValue <= 0)
        return;

    return Math.log(nValue);
}

function calc_PIP(aFirstCoord, aSecondCoord, xSource){

    var nIndexOfFirst = aFirstCoord[0];
    var nIndexOfSecond = aSecondCoord[0];
    var nCountOfBars = Math.abs(nIndexOfFirst - nIndexOfSecond) + 1;
```

```
    var nPrevMaxDiff = 0;
    var nPrevMaxIndex  = 0;

    var aMaxValue = [];

    for (var i = 0; i < nCountOfBars; i++){

        var nIndicatorValue = xSource.getValue(nIndexOfFirst + i);
        if (nIndicatorValue == null)
            return;

        var nCrossPoint = (((i * (aSecondCoord[1] - aFirstCoord[1])) / (nCountOfBars
- 1)) + aFirstCoord[1]);
        var nDiff = Math.abs(nCrossPoint - nIndicatorValue);

        aMaxValue = [];
        aMaxValue[0] = Math.max(nDiff, nPrevMaxDiff);
        aMaxValue[1] = aMaxValue[0] == nDiff ? nIndexOfFirst + i : nPrevMaxIndex;

        nPrevMaxDiff = aMaxValue[0];
        nPrevMaxIndex  = aMaxValue[1];
    }
    return aMaxValue;
}

function verify(){

    var b = false;
    if (getBuildNumber() < 779){

        drawTextAbsolute(5, 35, "This study requires version 8.0 or later.",
            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,
            null, 13, "error");
        drawTextAbsolute(5, 20, "Click HERE to upgrade.@URL=http://www.esignal.com
/download/default.asp",
            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,
            null, 13, "upgrade");
        return b;
    }
    else
        b = true;

    return b;
}
```

—Eric Lippert
eSignal, an Interactive Data company
800 779-6555, www.eSignal.com

BACK TO LIST

THINKORSWIM: MAY 2015

In “Filtering Price Movement” in this issue, author Giorgos Siligardos takes a fresh look at the old technical analysis tool zigzag. He discusses the limitations of a traditional zigzag indicator and introduces concepts to strengthen it.

We have recreated his zzTOP study using our proprietary scripting language *thinkscript*. We have made the loading process extremely easy: Simply click on the link http://tos.mx/npjeNL and choose “save script to thinkorswim,” then choose to rename your study to “zzTOP.” You can adjust the parameters of these within the *edit studies* window to fine-tune your variables.

In the example in Figure 3, we have added a 10-leg strategy version of the zzTOP on a thinkorswim chart of Ball Corp. (BLL). Refer to Siligardos’ article for a detailed description of the strategy.

FIGURE 3: THINKORSWIM. This example chart shows a 10-leg version of the zzTOP study on a daily chart of Ball Corp. (BLL).

—thinkorswim
A division of TD Ameritrade, Inc.
www.thinkorswim.com

BACK TO LIST

WEALTH-LAB: MAY 2015

We've implemented the *perceptually important points* (PIPs) method introduced by Giorgos Siligardos in his article in this issue, "Filtering Price Movement," in a script study. The script uses a recursive call to find the PIP having the maximum absolute value of the vertical distance from the line connecting tw o PIPs previously found. The price movement plot is based on a user-specified percentage.

As suggested by the article's author, for a DataSeries plotted in the *arithmetic* scale, the minimum vertical distance required to find a PIP is the percentage of the DataSeries' entire range, w hereas a fixed vertical distance in a *logarithmic* plot is inherently represented equally by the same percentage. For example, on a log chart, the distance betw een 1 and 10 is the same as that betw een 10 and 100 (or for any other 1,000% price change).

Finally, note that due to the manner in w hich the indicator is constructed, the zzTOP indicator *must not* be used for backtesting, but rather could be useful for digitally scanning numerous charts for patterns.

The Wealth-Lab code listing is show n below.

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Drawing;
using WealthLab;
using WealthLab.Indicators;
using Community.Components;

namespace WealthLab.Strategies
{
   internal struct PipInterval
   {
      public int X1;
      public int X2;
   }

   public class zzTOPAutoStudy : WealthScript
   {
      StrategyParameter _pct;
      StrategyParameter _thickness;
      List<int> _pipList;      // list of PIP bar numbers

      public zzTOPAutoStudy()
      {
         _pct = CreateParameter("PIP Percent", 20, 2, 50, 1);
         _thickness = CreateParameter("Line Width", 2, 1, 3, 1);
      }

      int getPipBar(DataSeries ds, PipInterval pi, double minMove, bool useLog)
      {
         int pip = 0;
         double maxDiff = 0;
         double delta, y;

         for(int bar = pi.X1; bar <= pi.X2; bar++)
         {
            if (useLog)
            {
               y = LineExtendYLog( pi.X1, ds[pi.X1], pi.X2, ds[pi.X2], bar );
```

```
            delta = Math.Abs(Math.Log(ds[bar] / y));
         }
         else
         {
            y = LineExtendY( pi.X1, ds[pi.X1], pi.X2, ds[pi.X2], bar );
            delta = Math.Abs(ds[bar] - y);
         }

         if( delta > maxDiff )
         {
            maxDiff = delta;
            pip = bar;
         }
      }

      if (maxDiff < minMove)
         pip = -1;   // invalid; no PIP in specified PipInterval

      if (pip > 0)
         _pipList.Add(pip);

      return pip;
   }

   /* Find the new PIP for each PipInterval in the List and return the new
PipIntervals */
   internal List<PipInterval> zzTopAuto(List<PipInterval> piList, DataSeries ds,
double minmove, bool useLog)
   {
      List<PipInterval> nextList = new List<PipInterval>();

      foreach (PipInterval pi in piList)
      {
         int bar = getPipBar(ds, pi, minmove, useLog);

         if (bar == -1)
            continue;
         else
         {
            PipInterval newinvl = new PipInterval();
            newinvl.X1 = pi.X1;
            newinvl.X2 = bar;
            nextList.Add(newinvl);

            PipInterval newinvl2 = new PipInterval();
            newinvl2.X1 = bar;
            newinvl2.X2 = pi.X2;
            nextList.Add(newinvl2);
         }
      }

      if (nextList.Count != 0)
         zzTopAuto(nextList, ds, minmove, useLog);

      return nextList;
   }

   void ZZTOP(DataSeries ds, ChartPane cp, double minPercent)
   {
      // Minimum vertical move for the arithmetic and log cases
```

```
         double minV = 0;
         if (cp.LogScale)
            minV = Math.Log(1 + minPercent/100d);
         else
         {
            int bc = Bars.Count;
            minV = minPercent / 100d * (Highest.Value(bc-1, ds, bc) -
Lowest.Value(bc-1, ds, bc));
         }

         // Initialize _pipList with the first and last bar numbers
         int nbars = Bars.Count - 1;
         _pipList = new List<int>();
         _pipList.Add(0);
         _pipList.Add(nbars);

         // Initialize the first list to pass to zzTopAuto
         PipInterval interval = new PipInterval();
         interval.X1 = 0;
         interval.X2 = nbars;
         List<PipInterval> aList = new List<PipInterval>();
         aList.Add(interval);

         // Let the recursion begin!
         zzTopAuto(aList, ds, minV, cp.LogScale);

         // Sort the result to plot lines between the PIPs
         _pipList.Sort();

         int lastpip = 0;
         foreach (int pip in _pipList)
         {
            if (pip == 0) continue;
            DrawLine(cp, lastpip, ds[lastpip], pip, ds[pip], Color.Blue,
LineStyle.Solid, _thickness.ValueInt);
            lastpip = pip;
         }

         int segments = _pipList.Count - 1;
         DrawLabel(cp, "Log Scale: " + cp.LogScale.ToString());
         DrawLabel(cp, "Segment Count: " + segments.ToString());
         DrawLabel(cp, "PIP: " + minPercent + "%");
         //DrawLabel(cp, "Press Go! after switch between Log and Arithmetic scales!",
Color.Red);
      }


      protected override void Execute()
      {
         ZZTOP(Close, PricePane, _pct.Value);

         DataSeries rsi = RSI.Series(Close, 14);
         ChartPane rsiPane = CreatePane(40, true, true);
         PlotSeries(rsiPane, rsi, Color.Black, LineStyle.Solid, 1);
         ZZTOP(rsi, rsiPane, _pct.Value);
      }
   }
}
```

See Figure 4 for an example chart.

Figure 4: WEALTH-LAB. Wealth-Lab's zzTOPauto routine automatically adjusts for the chart pane's log or arithmetic scale.

—Robert Sucher
Wealth-Lab, www.wealth-lab.com

BACK TO LIST

AMIBROKER: MAY 2015

In "Filtering Price Movement" in this issue, author Giorgos Siligardos presents a new zigzag-style indicator for visual identification of price patterns, which he calls zzTOP. Using AmiBroker's formula language (AFL), it's possible to write code directly in AmiBroker in a short and concise way without having to use an external DLL and/or external languages. The code listing follows.

LISTING 1.

```
SetBarsRequired( sbrAll, 0 );

function FindMiddlePIP( data, x0, x1, curdist )
```

```
{
  bi = BarIndex();

  y0 = data[ x0 ];
  y1 = data[ x1 ];

  line = y0 + ( y1 - y0 ) * ( bi - x0 ) / ( x1 - x0 );

  distance = abs(  data - line );
  maxbars = HHVBars( distance, x1 - x0 );
  pipbar = Nz( x1 - maxbars[ x1 ], -1 );

  if ( pipbar != -1 )
     VarSet( curdist, distance[ pipbar ] );

  return pipbar;
}

LogMode = ParamToggle( "Mode", "Linear|Logarithmic", 0 );
MaxLegs = Param( "MaxLegs", 20, 4, 35 );

data = Close;
// one leg first - from start to end
legs = 1;
pips = 0;
pips[ 0 ] = 1;
pips[ BarCount - 1 ] = 1;

if ( LogMode ) data = log( Data );

for ( ; legs < maxlegs; legs++ )
{
 x0 = 0;
 x1 = -1;
 oldpip = -1;
 maxdist = 0;

 for ( i = 0; i < BarCount; i++ )
 {
  if ( pips[ i ] )
  {
    x1 = i;

    if ( x1 > x0 )
    {
      curdist = 0;
      newpip = FindMiddlePIP( data, x0, x1, "curdist" );

      if ( newpip != -1 AND curdist > maxdist  )
      {
          maxdist = curdist;

          if ( oldpip != -1 )
              pips[ oldpip ] = 0; // remove smaller one

          pips[ newpip ] = 1;
          oldpip = newpip;
      }

      x0 = x1; // next leg
```

```
        x1 = -1; //
    }
   }
  }
 }

// draw lines connecting pip points
x0 = 0;
x1 = -1;
zzline = Null;
for ( i = 0; i < BarCount; i++ )
{
     if ( pips[ i ] )
     {
         x1 = i;
         line = LineArray( x0, data[ x0 ], x1, data[ x1 ] );
         zzline = IIf( NOT IsNull( line ), line, zzline );
         x0 = x1;
     }
}

if ( LogMode )
     zzline = exp( zzline );

Plot( zzline, "zzTop", colorRed, styleThick );
Plot( C, "Price", colorDefault );
```

An example of the zzTOP indicator as implemented on an AmiBroker chart of ASML Holdings is shown in Figure 5.

FIGURE 5: AMIBROKER. Here is a sample chart showing the zzTop indicator applied on ASML daily data, replicating the chart from Siligardos' article in this issue.

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

BACK TO LIST

NEUROSHELL TRADER: MAY 2015

The PIPs (*perceptually important points*) method described by Giorgos Siligardos in his article in this issue, "Filtering Price Movement," can be easily implemented in NeuroShell Trader using NeuroShell Trader's ability to call external dynamic linked libraries (DLLs). Dynamic linked libraries can be w ritten in C, C++, Pow er Basic, or Delphi.

After w riting the indicator code in your preferred compiler and creating a DLL, you can insert the resulting indicators as follow s:

1. Select *new indicator* from the insert menu
2. Choose the *External Program & Library Calls* category
3. Select the appropriate External DLL Call indicator
4. Set up the parameters to match your DLL
5. Select the *finished* button.

As noted by Siligardos in his article, these indicators "change their historical values w hen new data comes in" and thus should *not* be used for backtesting or automated trading, and instead should only be used "as a digital substitution for your eyes" w hen scanning charts.

Users of NeuroShell Trader can go to the STOCKS & COMMODITIES section of the NeuroShell Trader free technical support w ebsite to dow nload a copy of this or any previous Traders' Tips.

A sample chart is show n in Figure 6.

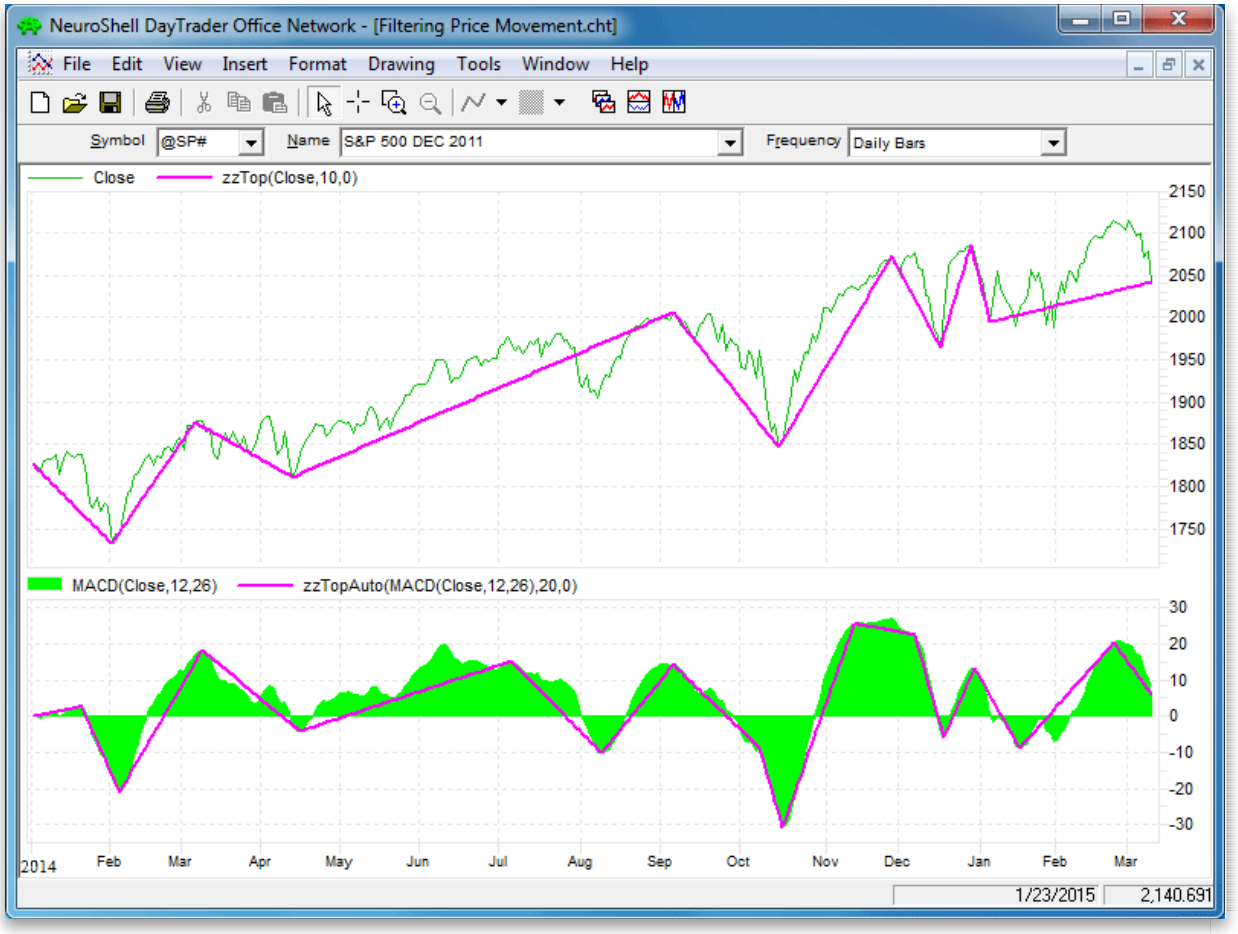


FIGURE 6: NEUROSHELL TRADER. This NeuroShell Trader chart shows the zzTOP indicator applied to the closing price and the zzTOPauto indicator applied to an MACD indicator.

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

BACK TO LIST

MICROSOFT EXCEL: MAY 2015

In "Filtering Price Movement" in this issue, author Giorgos Siligardos shows us a tool that lets us determine the level of price swing detail that we wish to pay attention to.

Much like zooming in on a map from satellite level down to neighborhood level, the closer we get, the more fine detail we can see. What he calls *perceptually important points* (PIPs) can do that for us. When we ask for only a few PIPs, we are highlighting the large (high-level) moves and ignoring the small ones. As we ask for more points, we are, in effect, zooming in to highlight the ever-finer details.

The zigzag indicator I will use here was excerpted from the spreadsheet I built for my June 2013 Traders' Tips submission (which readers can find in the Traders' Tips archive at Traders.com). I am using that indicator as a starting point since Siligardos' article in this issue compares the behaviors of his zzTOP indicators to the traditional zigzag.

My implemention of the zzTOP indicator in Excel has three user controls (see Figure 7):

- Type in the number of legs you want to see
- Click on the checkbox to toggle between arithmetic or logarithmic calculation modes
- Click one or more times on the gray button to select the pricing column to use in calculation of the indicator. This “tumbler” button includes a hybrid choice of *Hi:Lo Combo*.

In the *Hi:Lo Combo* mode, both the high and low of a bar are tested when selecting candidate points. This behavior is similar in concept to the standard zigzag and can produce a very choppy zzTOP indicator.

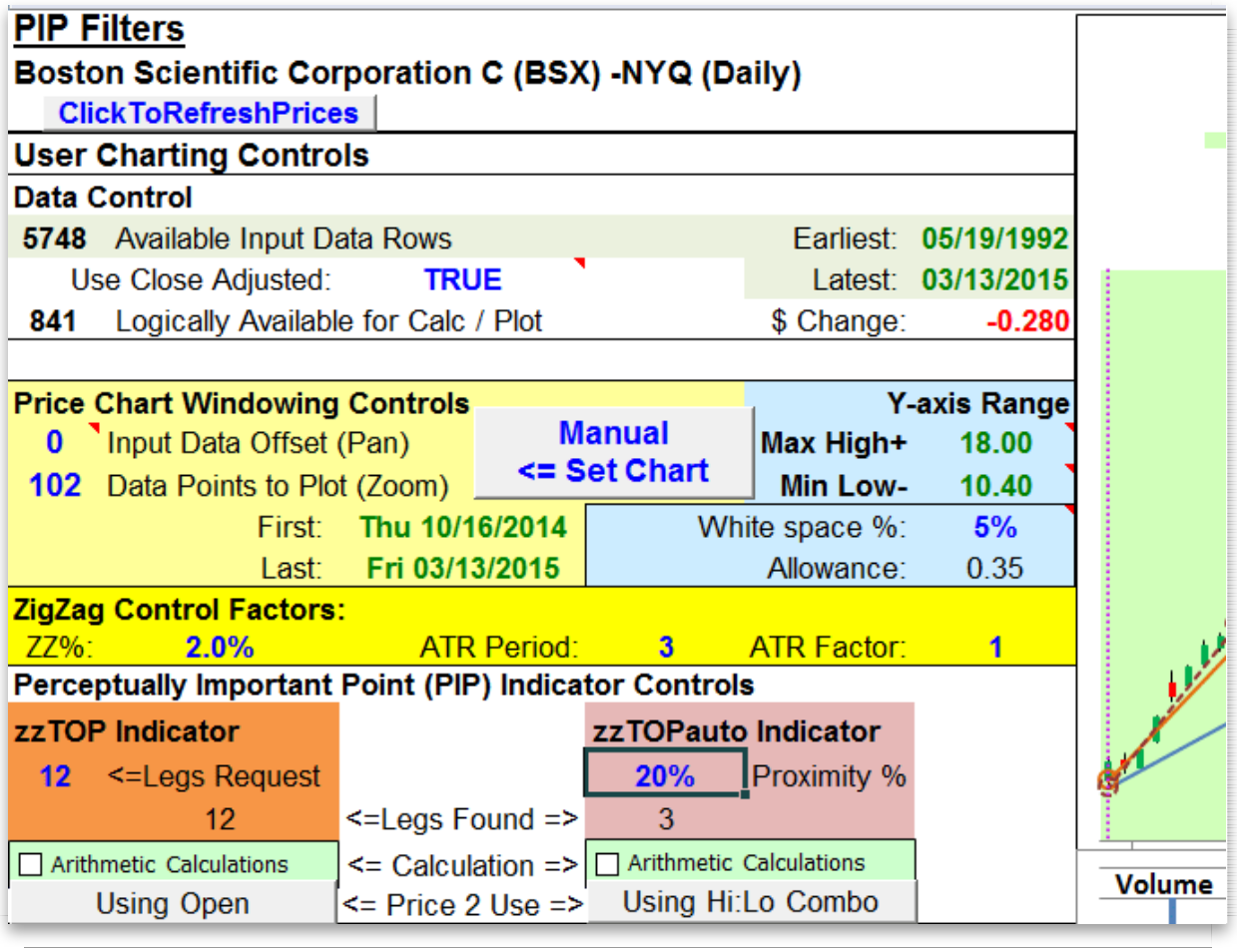


FIGURE 7: EXCEL, User Controls. Here you see the controls implemented in the spreadsheet for the zigzag, zzTOP, and zzTOPauto indicators.

The zzTOPauto version of the indicator swaps the specific number of legs for a proximity test, but otherwise, the controls behave the same way.

To be able to see the effects of changing the starting and ending points for the zzTOP indicators, I have provided the ability to specify the interval of interest via slider controls (Figure 8). These can be used to step the left and right edges of the computation interval in or out and immediately see what happens to the indicator.

FIGURE 8: EXCEL, zzTOP Interval Controls. Slider controls allow you to immediately see the effects of changing the starting and ending points for the zzTOP indicator.

Use the checkboxes to select the indicator or indicators you w ish to display as a w ay to control chart clutter.

In Figure 9, the interval is the full chart and w e can see how the zzTOP indicators stack up against a standard zigzag over the same interval.

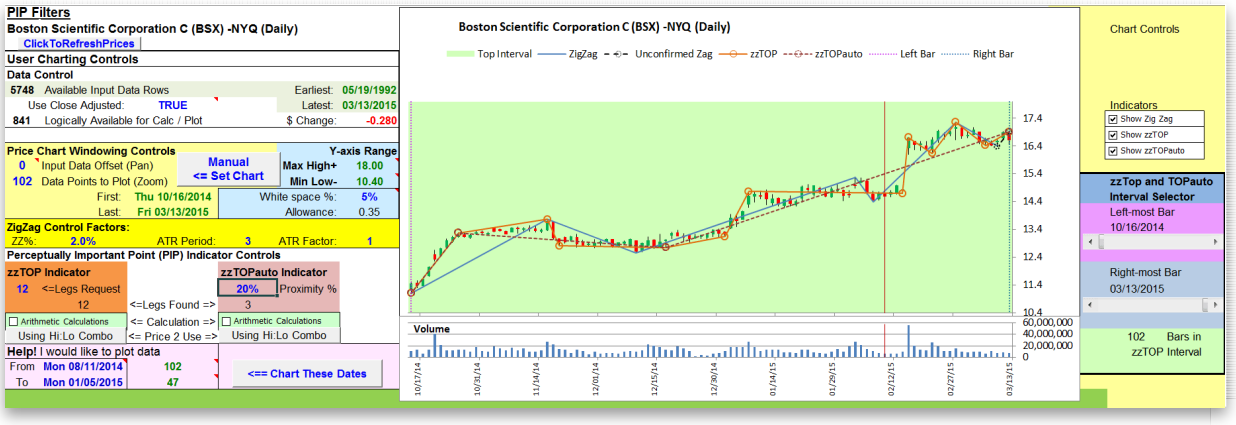


FIGURE 9: EXCEL, ZigZag and All. Here, the interval is the full chart and we can see how the zzTOP indicators stack up against a standard zigzag over the same interval.

ADDITIONAL USES FOR PIPS (PERCEPTUALLY IMPORTANT POINTS)

In 2010, w hile Google and I w ere looking for discussions of pattern matching in time series data, I came across an interesting master's thesis w ritten in 2008 titled "Novel Pattern Matching Methods For Stock Data Analysis" by Zhang Zhe, City University of Hong Kong. It's a fairly accessible read and proposes a

three-step process for locating and identifying patterns such as head & shoulders in a time series stream. Interested readers can access a free PDF of the paper at http://lbms03.cityu.edu.hk/theses/abt/mphil-is-b23405983a.pdf.

The spreadsheet file for this Traders' Tip can be downloaded here. To successfully download it, follow these steps:

- Right-click on the Excel file link, then
- Select "save as" (or "save target as") to place a copy of the spreadsheet file on your hard drive.

—Ron McAllister
Excel and VBA programmer
rpmac_xltt@sprynet.com

BACK TO LIST

June 2015

For this month's Traders' Tips, the focus is Kevin Luo's article in this issue, "The RSI & Price Trends." Here, we present the June 2015 Traders' Tips code with possible implementations in various software.

The Traders' Tips section is provided to help the reader implement a selected technique from an article in this issue or another recent issue. The entries here are contributed by software developers or programmers for software that is capable of customization.

TRADESTATION: JUNE 2015

In "The RSI & Price Trends" in this issue, author Kevin Luo describes several new approaches to trading with the well-known relative strength index (RSI) indicator. One of the author's approaches is to only trade with the prevailing trend. He has defined a trend change as occurring when there is a 20% retracement from the extreme price of the current trend. A second approach Luo describes is to only select stocks for trading that exhibit shorter trend lengths.

For convenience, we are providing code for a TradeStation strategy based on the author's ideas. In addition to the strategy code, we are providing two accompanying indicators. The first, *TrendInfo*, can be placed in the chart along with the strategy to assist in visualizing the trends and trend changes. The second, *TrendLength*, is intended to be used in the TradeStation Scanner to assist in identifying average trend length of a list of stocks. From the Scanner results, a symbol list can be generated and then backtested as a portfolio using the TradeStation Portfolio Maestro application.

To download the EasyLanguage code, please visit our TradeStation and EasyLanguage support forum. The code for this article can be found here: http://www.tradestation.com/TASC-2015. The ELD filename is "TASC_JUN2015.ELD." The code is also shown here:

```
Strategy: RSI with Trend

// Tasc Jun 2015
```

```
// RSI and Price Trends
// Strategy


inputs:
        Price( Close ) ,
        RSILength( 14 ),
        OverBought( 70 ),
        Oversold( 30 ),
        RetracePct( 20 ),
        ExitOnTrendChange( true ),
        UseTrendFilter( true ),
        DrawTrendLines( true ),
        LineColor( Yellow ),
        LineWidth( 1 );

variables:
        NewSwingPrice( 0 ),
        SwingPrice( Price ),
        SwingDate( Date ),
        SwingTime( Time ),
        TLDir( 0 ),
        RetraceFctrUp( 1 + RetracePct * .01 ),
        RetraceFctrDn( 1 - RetracePct * .01 ),
        SaveSwing( false ),
        AddTL( false ),
        UpdateTL( false ),
        TLRef( 0 ),
        RSIValue( 0 ),
        TrendOK( false ) ;

NewSwingPrice = SwingHigh( 1, Price, 1, 2 ) ;

if NewSwingPrice  <> -1 then
        begin
        if TLDir <= 0 and NewSwingPrice >= SwingPrice
                * RetraceFctrUp then
                begin
                SaveSwing = true ;
                AddTL = true ;
                TLDir = 1 ;
                end
        else if TLDir = 1 and NewSwingPrice >= SwingPrice then
                begin
                SaveSwing = true ;
                UpdateTL = true ;
                end ;
        end
else
        begin
        NewSwingPrice = SwingLow( 1, Price, 1, 2 ) ;
        if NewSwingPrice <> -1 then
                begin
                if TLDir >= 0 and NewSwingPrice <= SwingPrice
                        * RetraceFctrDn then
                        begin
                        SaveSwing = true ;
                        AddTL = true ;
                        TLDir = -1 ;
                        end
```

```
                else if TLDir = -1 and NewSwingPrice <= SwingPrice then
                        begin
                        SaveSwing = true;
                        UpdateTL = true ;
                        end ;
                end ;
        end ;

if SaveSwing then
        begin
        SwingPrice = NewSwingPrice ;
        SwingDate = Date[1] ;
        SwingTime = Time[1] ;
        SaveSwing = false ;
        end ;

if DrawTrendLines = true then
        begin
        if AddTL then
                begin
                TLRef = TL_New( SwingDate, SwingTime, SwingPrice,
                SwingDate[1], SwingTime[1], SwingPrice[1] ) ;
                TL_SetExtLeft( TLRef, false ) ;
                TL_SetExtRight( TLRef, false ) ;
                TL_SetSize( TLRef, LineWidth ) ;
                TL_SetColor( TLRef, LineColor ) ;
                AddTL = false ;
                end
        else if UpdateTL then
                begin
                TL_SetEnd( TLRef, SwingDate, SwingTime, SwingPrice ) ;
                UpdateTL = false ;
                end ;
        end ;

RSIValue = RSI( Price, RSILength ) ;

if UseTrendFilter = false then
        TrendOK = true
else
        TrendOK = TLDir = 1 ;

if RSIValue crosses under OverSold and TrendOK then
        Buy ( "RSI LE" )next bar at Market
else if RSIValue crosses over OverBought then
        Sell ( "RSI LX" ) next bar at Market ;

if ExitOnTrendChange and TLDir = -1 then
        Sell ( "Trend LX" ) next bar at Market ;
```

Indicator: TrendInfo

```
// Tasc Jun 2015
// RSI and Price Trends
// Chart indicator used to show trend changes
// and trend statistics
```

```
using elsystem ;
using elsystem.collections ;

inputs:
        Price( Close ) ,
        RetracePct( 20 ) ,
        LineColor( Yellow ),
        LineWidth( 1 ),
        DrawTrendLines( true )  ;

variables:
        NewSwingPrice( 0 ),
        SwingPrice( Price ),
        SwingDate( Date ),
        SwingTime( Time ),
        TLDir( 0 ),
        RetraceFctrUp( 1 + RetracePct * .01 ),
        RetraceFctrDn( 1 - RetracePct * .01 ),
        SaveSwing( false ),
        AddTL( false ),
        UpdateTL( false ),
        TLRef( 0 ),
        AvgTrendLength( 0 ),
        LastTrendChgBar( 0 ),
        vector TrendLength( NULL ) ;

once
        begin
        TrendLength = new Vector ;
        end ;

NewSwingPrice = SwingHigh( 1, Price, 1, 2 ) ;

if NewSwingPrice  <> -1 then
        begin
        if TLDir <= 0 and NewSwingPrice >= SwingPrice
                * RetraceFctrUp then
                begin
                SaveSwing = true ;
                AddTL = true ;
                TLDir = 1 ;
                end
        else if TLDir = 1 and NewSwingPrice >= SwingPrice then
                begin
                SaveSwing = true ;
                UpdateTL = true ;
                end ;
        end
else
        begin
        NewSwingPrice = SwingLow( 1, Price, 1, 2 ) ;
        if NewSwingPrice <> -1 then
                begin
                if TLDir >= 0 and NewSwingPrice <= SwingPrice
                        * RetraceFctrDn then
                        begin
                        SaveSwing = true ;
                        AddTL = true ;
                        TLDir = -1 ;
                        end
```

```
                else if TLDir = -1 and NewSwingPrice <= SwingPrice then
                        begin
                        SaveSwing = true;
                        UpdateTL = true ;
                        end ;
                end ;
        end ;

if SaveSwing then
        begin
        SwingPrice = NewSwingPrice ;
        SwingDate = Date[1] ;
        SwingTime = Time[1] ;
        SaveSwing = false ;
        end ;

if DrawTrendLines = true then
        begin
        if AddTL then
                begin
                TLRef = TL_New( SwingDate, SwingTime, SwingPrice,
                SwingDate[1], SwingTime[1], SwingPrice[1] ) ;
                TL_SetExtLeft( TLRef, false ) ;
                TL_SetExtRight( TLRef, false ) ;
                TL_SetSize( TLRef, LineWidth ) ;
                TL_SetColor( TLRef, LineColor ) ;
                AddTL = false ;
                end
        else if UpdateTL then
                begin
                TL_SetEnd( TLRef, SwingDate, SwingTime, SwingPrice ) ;
                UpdateTL = false ;
                end ;
        end ;

if LastTrendChgBar = 0 and TLDir <> TLDir[1] then
        begin
        LastTrendChgBar = CurrentBar ;
        TrendLength.push_back( LastTrendChgBar astype int ) ;
        end
else if TLDir <> TLDir[1] then
        begin
        TrendLength.push_back( CurrentBar - LastTrendChgBar astype int ) ;
        LastTrendChgBar = CurrentBar ;
        end ;

if LastBarOnChartEx and TrendLength.Count > 0 then
        AvgTrendLength = Average( TrendLength, TrendLength.Count ) ;

Plot1( TLDir ) ;
Plot2( " AvgTrendLength : " +  NumToStr( AvgTrendLength, 0 )
                + " Bars", "AvgTrendLen", Yellow ) ;
Plot3( " Num Trends : " + NumToStr( TrendLength.Count, 0 ),
                "Num Trends",Yellow ) ;
```

Indicator: TrendLength

```
// Tasc Jun 2015
// RSI and Price Trends
```

```
// The indicator is intended to be used in scanner
// to calculate average trend length for each symbol
//    in a symbol list.
// Be sure to load additional data based on the period
//    that you want to analyze


using elsystem ;
using elsystem.collections ;

inputs:
        Price( Close ) ,
        RetracePct( 20 ) ;

variables:
        NewSwingPrice( 0 ),
        SwingPrice( Price ),
        SwingDate( Date ),
        SwingTime( Time ),
        TLDir( 0 ),
        RetraceFctrUp( 1 + RetracePct * .01 ),
        RetraceFctrDn( 1 - RetracePct * .01 ),
        SaveSwing( false ),
        AddTL( false ),
        UpdateTL( false ),
        TLRef( 0 ),
        AvgTrendLength( 0 ),
        LastTrendChgBar( 0 ),
        vector TrendLength( NULL ) ;

once
        begin
        TrendLength = new Vector ;
        end ;

NewSwingPrice = SwingHigh( 1, Price, 1, 2 ) ;

if NewSwingPrice  <> -1 then
        begin
        if TLDir <= 0 and NewSwingPrice >= SwingPrice
                * RetraceFctrUp then
                begin
                SaveSwing = true ;
                AddTL = true ;
                TLDir = 1 ;
                end
        else if TLDir = 1 and NewSwingPrice >= SwingPrice then
                begin
                SaveSwing = true ;
                UpdateTL = true ;
                end ;
        end
else
        begin
        NewSwingPrice = SwingLow( 1, Price, 1, 2 ) ;
        if NewSwingPrice <> -1 then
                begin
                if TLDir >= 0 and NewSwingPrice <= SwingPrice
                        * RetraceFctrDn then
                        begin
```

```
                        SaveSwing = true ;
                        AddTL = true ;
                        TLDir = -1 ;
                        end
                else if TLDir = -1 and NewSwingPrice <= SwingPrice then
                        begin
                        SaveSwing = true;
                        UpdateTL = true ;
                        end ;
                end ;
        end ;

if SaveSwing then
        begin
        SwingPrice = NewSwingPrice ;
        SwingDate = Date[1] ;
        SwingTime = Time[1] ;
        SaveSwing = false ;
        end ;

if LastTrendChgBar = 0 and TLDir <> TLDir[1] then
        begin
        LastTrendChgBar = CurrentBar ;
        TrendLength.push_back( LastTrendChgBar astype int ) ;
        end
else if TLDir <> TLDir[1] then
        begin
        TrendLength.push_back( CurrentBar - LastTrendChgBar astype int ) ;
        LastTrendChgBar = CurrentBar ;
        end ;

if LastBarOnChartEx and TrendLength.Count > 0 then
        AvgTrendLength = Average( TrendLength, TrendLength.Count ) ;

Plot1(  AvgTrendLength,  " TrendLen" ) ;
```

For more information about EasyLanguage in general, please see http://w w w .tradestation.com/EL-FAQ.

A sample chart is show n in Figure 1.

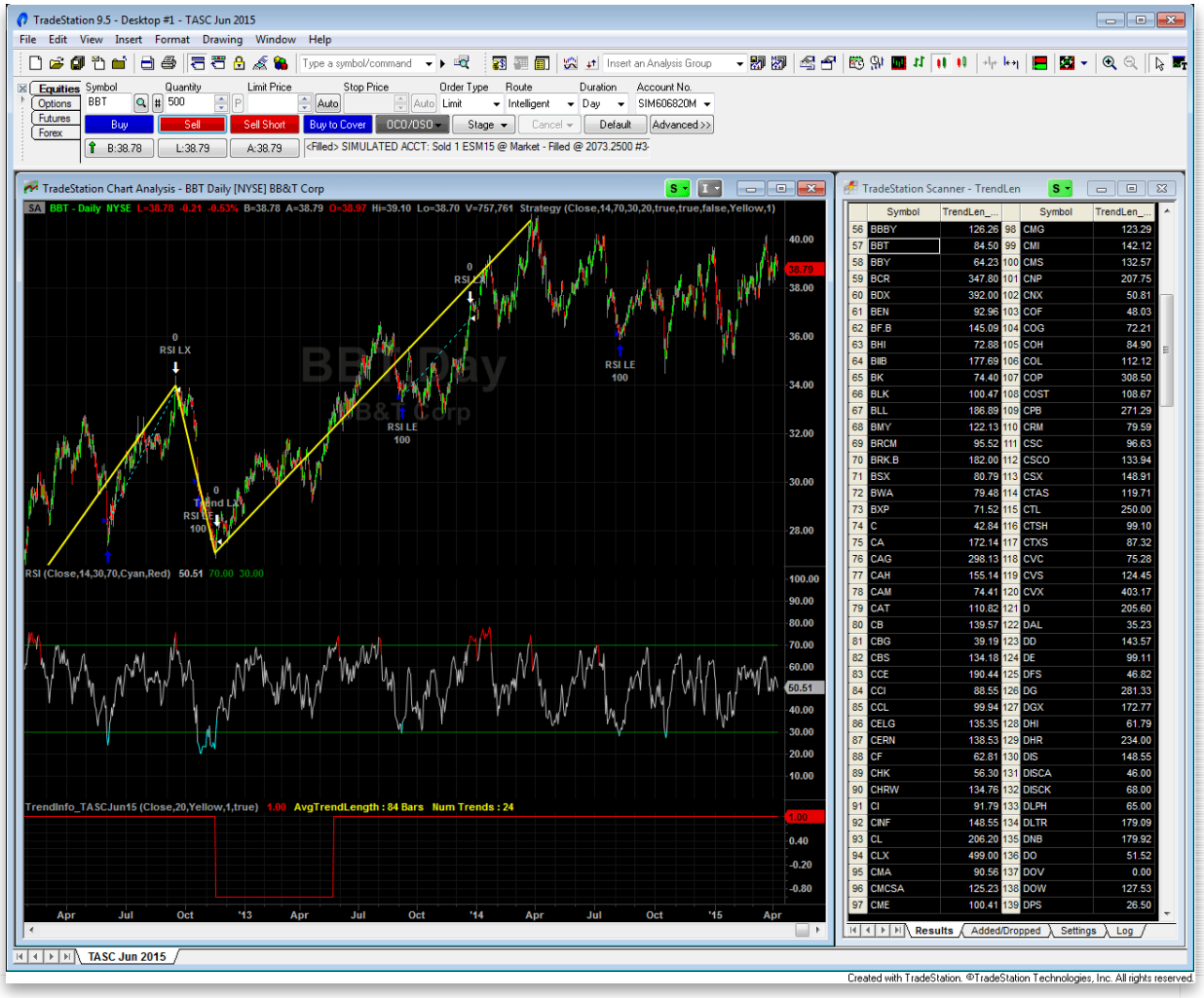


FIGURE 1: TRADESTATION, RSI WITH TREND. Here is an example of the custom RSI strategy and TrendInfo indicator applied to a daily chart of BBT along with the average TrendLength indicator as used with the TradeStation Scanner.

This article is for informational purposes. No type of trading or investment recommendation, advice, or strategy is being made, given, or in any manner provided by TradeStation Securities or its affiliates.

—Doug McCrary
TradeStation Securities, Inc.
www.TradeStation.com

BACK TO LIST

eSIGNAL: JUNE 2015

For this month's Traders' Tip, w e're providing the formula StrategyRSI.efs based on the formula described in Kevin Luo's article in this issue, "The RSI & Price Trends."

This formula takes long trades only and is back-testable by using the *show backtest* feature in the program.

The study contains formula parameters that may be configured through the *edit chart* window (right-click on the chart and select "edit chart"). A sample chart implementing this study is shown in Figure 2.

FIGURE 2: eSIGNAL. Here is an example of the study implemented on a chart of Lennar Corp. (LEN), based on Kevin Luo's article in this issue.

To discuss this study or download a complete copy of the formula code, please visit the EFS Library Discussion Board forum under the *forums* link from the support menu at www.esignal.com or visit our EFS KnowledgeBase at http://www.esignal.com/support/kb/efs/. The eSignal formula script (EFS) is also available here:

```
/**********************************

Provided By:

    Interactive Data Corporation (Copyright -í© 2015)

    All rights reserved. This sample eSignal Formula Script (EFS)

    is for educational purposes only. Interactive Data Corporation

    reserves the right to modify and overwrite this EFS file with
```

```
    each new release.


Description:

    The RSI & Price Trends


Formula Parameters:                     Default:

Length RSI                              14

Price Change, %                         20

RSI Upper Bound                         70

RSI Lower Bound                         50

Confirmed Long Zone                     lime

Long Trend Color                        green

Short Trend Color                       red

Long Position Color                     green

Short Position Color                    red

Nick Color                              grey

Nick Extension                          3

Nick Thickness                          3


Version:            1.00  04/09/2015


Notes:

    The related article is copyrighted material. If you are not a subscriber

    of Stocks & Commodities, please visit www.traders.com.


**********************************/


var fpArray = new Array();


function preMain(){
```

```
setStudyTitle("StrategyRSI");
setPriceStudy(true);
setColorPriceBars(true);
setComputeOnClose(true);

var x = 0;

fpArray[x] = new FunctionParameter("fpLength", FunctionParameter.NUMBER);
with(fpArray[x++]){

    setName("Length RSI");
    setLowerLimit(1);
    setDefault(14);
};

fpArray[x] = new FunctionParameter("fpChange", FunctionParameter.NUMBER);
with(fpArray[x++]){

    setName("Price Change, %");
    setLowerLimit(0);
    setUpperLimit(100);
    setDefault(20);
};

fpArray[x] = new FunctionParameter("fpRSIHighBorder", FunctionParameter.NUMBER);
with(fpArray[x++]){

    setName("RSI Upper Bound");
    setLowerLimit(0);
```

```
    setUpperLimit(100);
    setDefault(70);
};

fpArray[x] = new FunctionParameter("fpRSILowBorder", FunctionParameter.NUMBER);
with(fpArray[x++]){

    setName("RSI Lower Bound");
    setLowerLimit(0);
    setUpperLimit(100);
    setDefault(50);
};

fpArray[x] = new FunctionParameter("fpZoneColor", FunctionParameter.COLOR);
with(fpArray[x++]){

    setName("Confirmed Long Zone");
    setDefault(Color.lime);
};

fpArray[x] = new FunctionParameter("fpLongTrendColor", FunctionParameter.COLOR);
with(fpArray[x++]){

    setName("Long Trend Color");
    setDefault(Color.green);
};

fpArray[x] = new FunctionParameter("fpShortTrendColor", FunctionParameter.COLOR);
with(fpArray[x++]){
```

```
    setName("Short Trend Color");
    setDefault(Color.red);
};

fpArray[x] = new FunctionParameter("fpLongPosColor", FunctionParameter.COLOR);
with(fpArray[x++]){

    setName("Long Position Color");
    setDefault(Color.green);
};

fpArray[x] = new FunctionParameter("fpShortPosColor", FunctionParameter.COLOR);
with(fpArray[x++]){

    setName("Short Position Color");
    setDefault(Color.red);
};

fpArray[x] = new FunctionParameter("fpNickColor", FunctionParameter.COLOR);
with(fpArray[x++]){

    setName("Nick Color");
    setDefault(Color.grey);
};

fpArray[x] = new FunctionParameter("fpExtNick", FunctionParameter.NUMBER);
with(fpArray[x++]){

    setName("Nick Extension");
    setLowerLimit(0);
```

```
        setDefault(3);
    };

    fpArray[x] = new FunctionParameter("fpThicNick", FunctionParameter.NUMBER);
    with(fpArray[x++]){

        setName("Nick Thickness");
        setLowerLimit(1);
        setUpperLimit(10)
        setDefault(3);
    };
}

var bInit = false;
var bVersion = null;

var xOpen = null;
var xHigh = null;
var xLow = null;

var xRSI = null;

var nChange = null;
var nHighLim = null;
var nLowLim = null;
var nDefLotSize = null;

var nTagID = 0;

var bIsTrend = false;
```

```
var bLongTrend = false;

var bShortTrend = false;

var aHignest = {};

var aLowest = {};


function main(fpLength, fpChange,

              fpRSIHighBorder, fpRSILowBorder,

              fpLongTrendColor, fpShortTrendColor,

              fpZoneColor, fpLongPosColor, fpShortPosColor,

              fpNickColor, fpExtNick, fpThicNick){


    if (bVersion == null) bVersion = verify();

    if (bVersion == false) return;


    if (!bInit){


        xOpen = open();

        xHigh = high();

        xLow = low();


        xRSI = rsi(fpLength);


        nChange = fpChange / 100;

        nHighLim = 1 + nChange;

        nLowLim = 1 - nChange;


        nDefLotSize = Strategy.getDefaultLotSize();


        bInit = true;

    };
```

```
if (getBarState() == BARSTATE_ALLBARS){
    nTagID = 0;
    bIsTrend = false;
    bLongTrend = false;
    bShortTrend = false;
    aHighest = {};
    aLowest = {};
}

var bReverseThisBar = false;

var nNextOpen = xOpen.getValue(1);
var nOpen = xOpen.getValue(0);
var nHigh = xHigh.getValue(0);
var nLow = xLow.getValue(0);
var nPrevHigh = xHigh.getValue(-1);
var nPrevLow = xLow.getValue(-1);

var nRSI = xRSI.getValue(0);

if (nPrevHigh == null || nPrevLow == null)
    return;

var nCurrentBarNumber = getCurrentBarCount();

if (bShortTrend || !bIsTrend){

    if (nPrevLow < aLowest[0] || aLowest[0] == null){
        aLowest[0] = nPrevLow;
```

```
            aLowest[1] = nCurrentBarNumber - 1;
        }

        if (bIsTrend) setPriceBarColor(fpShortTrendColor);

        if (nHigh >= nHighLim * aLowest[0]){

            recalculate(nCurrentBarNumber - aLowest[1] + 1, function(){
                setPriceBarColor(fpLongTrendColor);
            })

            drawLineRelative(-fpExtNick, nHighLim * aLowest[0], fpExtNick, nHighLim *
aLowest[0],
                              null, fpThicNick, fpNickColor, nTagID++);

            if (!bIsTrend) bIsTrend = true;
            bLongTrend = true;
            bShortTrend = false;
            bReverseThisBar = true;
            aHighest[0] = nHigh;
            aHighest[1] = nCurrentBarNumber;
        }
    }

    if (bLongTrend || !bIsTrend){

        if (bIsTrend) setPriceBarColor(fpLongTrendColor);

        if ((nPrevHigh > aHighest[0] || aHighest[0] == null) && !bReverseThisBar ){
            aHighest[0] = nPrevHigh;
            aHighest[1] = nCurrentBarNumber - 1;
```

```
        }

        if (nLow <= nLowLim * aHighest[0] && !bReverseThisBar){

            recalculate(nCurrentBarNumber - aHighest[1] + 1, function(){
                setPriceBarColor(fpShortTrendColor);
            })

            if (bIsTrend) drawLineRelative(-fpExtNick, nLowLim * aHighest[0],
fpExtNick, nLowLim * aHighest[0],
                        null, fpThicNick, fpNickColor, nTagID++);

            if (!bIsTrend) bIsTrend = true;
            bLongTrend = false;
            bShortTrend = true;
            aLowest[0] = nLow;
            aLowest[1] = nCurrentBarNumber;

            var nExitPrice = Math.min(nOpen, nLowLim * aHighest[0]);

            if (Strategy.isLong()){
                Strategy.doSell("Exit Long", Strategy.LIMIT, Strategy.THISBAR,
Strategy.DEFAULT, nExitPrice);
                drawShapeRelative(0, AboveBar1, Shape.DOWNTRIANGLE, null,
fpShortPosColor,
                Text.PRESET, nTagID++);
                drawTextRelative(0, AboveBar2, "Exit Long", fpShortPosColor, null,
                Text.PRESET|Text.CENTER|Text.BOLD, null, null, nTagID++);
                drawTextRelative(0, AboveBar3, nDefLotSize + " @ " +
formatPriceNumber(nExitPrice), fpShortPosColor, null,
                Text.PRESET|Text.CENTER|Text.BOLD, null, null, nTagID++);
            }
```

```
        }

        else if (bIsTrend){



            setBarBgColor(fpZoneColor);



            if (nRSI == null)

                return;



            if (nRSI < fpRSILowBorder && !Strategy.isLong()){

                Strategy.doLong("Long", Strategy.MARKET, Strategy.NEXTBAR,
Strategy.DEFAULT);

                drawShapeRelative(1, BelowBar1, Shape.UPTRIANGLE, null,
fpLongPosColor,

                Text.PRESET, nTagID++);

                drawTextRelative(1, BelowBar2, "Long", fpLongPosColor, null,

                Text.PRESET|Text.CENTER|Text.BOLD, null, null, nTagID++);

                drawTextRelative(1, BelowBar3, nDefLotSize + " @ " +
formatPriceNumber(nNextOpen), fpLongPosColor, null,

                Text.PRESET|Text.CENTER|Text.BOLD, null, null, nTagID++);

            }



            if (nRSI > fpRSIHighBorder && Strategy.isLong()){

                Strategy.doSell("Exit Long", Strategy.MARKET, Strategy.NEXTBAR,
Strategy.DEFAULT);

                drawShapeRelative(1, AboveBar1, Shape.DOWNTRIANGLE, null,
fpShortPosColor,

                Text.PRESET, nTagID++);

                drawTextRelative(1, AboveBar2, "Exit Long", fpShortPosColor, null,

                Text.PRESET|Text.CENTER|Text.BOLD, null, null, nTagID++);

                drawTextRelative(1, AboveBar3, nDefLotSize + " @ " +
formatPriceNumber(nNextOpen), fpShortPosColor, null,

                Text.PRESET|Text.CENTER|Text.BOLD, null, null, nTagID++);

            }
```

```
        }

    }

}


function verify(){


    var b = false;

    if (getBuildNumber() < 3435){


        drawTextAbsolute(5, 35, "This study requires version 11.8 or later.",

            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,

            null, 13, "error");

        drawTextAbsolute(5, 20, "Click HERE to upgrade.@URL=http://www.esignal.com
/download/default.asp",

            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,

            null, 13, "upgrade");

        return b;

    }

    else

        b = true;



    return b;

}
```

—Eric Lippert
eSignal, an Interactive Data company
800 779-6555, www.eSignal.com

BACK TO LIST

Kevin Luo's article in this issue, "The RSI & Price Trends," discusses w ays to improve on the traditional RSI buy & sell signals. The MetaStock formulas given here are one possible method of using these ideas.

Buy Order:

```
peakbars(1, c, 20) > troughbars(1,c,20) AND
barssince(c>trough(1,c,20) < troughbars(1, c, 20) AND cross(RSI(14), 50)
```

Sell Order:

```
peakbars(1, c, 20) < troughbars(1,c,20) AND
Cross(troughbars(1, c, 20), C)
```

—William Golson
MetaStock Technical Support
www.metastock.com

BACK TO LIST

WEALTH-LAB: JUNE 2015

RSI is many a technical analyst's darlin', one of industry-standard indicators that has served as a playground for many modifications. In his article in this issue, "The RSI & Price Trends," Kevin Luo demonstrates that RSI classic countertrend trading rules can be traded profitably and w ithout modifications. According to the author, buying at the close w hen RSI falls below 30 and selling w hen it surges above 70 w orks.

Here is one caveat that should not be overlooked: Out of the 10-year backtest period, 80% of the time w as spent in a major bull market. In the last 10 years, "correctional movements," as the author calls them, of 2008, 2010, and 2011 in the Russell 2000 or S&P 500 indexes accounted for just tw o years in sum. In such a market, trading the corrections is natural and easy; but for how long w ill the bubble last?

To improve trading results, the author investigates modifying the RSI trading rules using a couple of logical and simple filters. First, it's possible to get rid of the less-probable trades by making entries only during uptrends. An uptrend is defined w ith a trailing filter that is activated w hen prices move up by more than 20% from a low (and vice versa for dow ntrend). This should help reduce the number of trades during a dow ntrend. Another tw eak is to close positions w hen a 20% uptrend reverses, that is, w hen prices decline 20% or greater from the last uptrend's peak.

Finally, trades can be taken w hen the RSI appears stronger and crosses above 50 rather than 30. From our observations, it's this rising RSI requirement that has the salient effect on the system's profitability.

FIGURE 3: WEALTH-LAB. This Wealth-Lab 6 chart illustrates the sample system's performance on a daily chart of American Express (AXP).

Figure 3 demonstrates the application of two filters: exiting a position when price falls below 20% from a trend's peak, and subsequent entries made in an uptrend when the RSI declined below 50. In the resulting RSI trading system written in C# for Wealth-Lab 6, all the options can be activated individually. The code is as follows:

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Drawing;
using WealthLab;
using WealthLab.Indicators;
using Community.Indicators;

namespace WealthLab.Strategies
{
        public class TASCJune2015 : WealthScript
        {
                private StrategyParameter paramRSI;
                private StrategyParameter paramMult;
                private StrategyParameter paramUseTF;
                private StrategyParameter paramUseRisingRSI;
                private StrategyParameter paramCloseOnReverse;

                public TASCJune2015()
                {
                        paramRSI = CreateParameter("RSI Period", 14, 2, 30, 2);
                        paramMult = CreateParameter("% Reversal", 20, 5, 50, 5);
                        paramUseTF = CreateParameter("Trend filter?", 0, 0, 1, 1);
                        paramUseRisingRSI = CreateParameter("Rising RSI?", 0, 0, 1,
1);
                        paramCloseOnReverse = CreateParameter("Close on reverse?", 0,
0, 1, 1);
          }

                protected override void Execute()
                {
```

```
                        double Mult = paramMult.Value;
                        bool trendFilter = paramUseTF.ValueInt == 1 ? true : false;

                        // Entering when the RSI is moving up and crossing above 50.
                        bool risingRSI = paramUseRisingRSI.ValueInt == 1 ? true :
false;

                        // Closing positions when uptrend reverses, that is, prices
move down by more than 20% from the high of prior uptrend.
            bool closeOnTrendReverse = paramCloseOnReverse.ValueInt == 1 ? true :
false;
                        double ob = 70, os = risingRSI ? 50 : 30;

                        NRTR_Percent trendLine = NRTR_Percent.Series( Bars, Mult );
                        //PlotSeries( PricePane, trendLine, Color.Blue,
LineStyle.Solid, 2 );

                        int rsiPeriod = paramRSI.ValueInt;
                        RSI rsi = RSI.Series( Close, rsiPeriod );
                        ChartPane rsiPane = CreatePane( 30,true,true );
                        PlotSeries( rsiPane, rsi, Color.Violet, LineStyle.Solid, 2 );
                        DrawHorzLine( rsiPane, ob, Color.Blue, LineStyle.Dashed, 1 );
                        DrawHorzLine( rsiPane, os, Color.Blue, LineStyle.Dashed, 1 );
            for(int bar = GetTradingLoopStartBar(20); bar < Bars.Count; bar++)
                        {
                                //      Entering trades only during uptrends, that
is, when prices move up by more than 20% from low of prior uptrend.
                                bool uptrend = Close[bar] > trendLine[bar];
                                SetBackgroundColor( bar, Color.FromArgb( 30, uptrend
? Color.Green : Color.Red ) );

                                if (IsLastPositionActive)
                                {
                                        if( closeOnTrendReverse )
                                        {
                                                if( !uptrend )
                                                        SellAtMarket(bar+1,
LastPosition, "Uptrend reversed" );
                                        }

                                        if( CrossOver( bar, rsi, ob ) )
                                        {
                                                SellAtMarket(bar+1, LastPosition,
"RSI" );
                                        }
                                }
                                else
                                {
                                        if( CrossUnder( bar, rsi, os ) )
                                        {
                                                if( ( trendFilter && uptrend ) ||
!trendFilter )
                                                        BuyAtMarket(bar+1,
Bars.FormatValue(rsi[bar]) );
                                        }
                                }
                        }
                }
        }
}
```

—Eugene, Wealth-Lab team
MS123, LLC
www.wealth-lab.com

BACK TO LIST

AMIBROKER: JUNE 2015

In "The RSI & Price Trends" in this issue, author Kevin Luo presents a simple trading system based on RSI signals filtered w ith a trend indicator. A ready-to-use formula based on Luo's system is show n here for AmiBroker. It can be applied either on a chart or in the analysis w indow for backtesting.

A sample chart is show n in Figure 4.

FIGURE 4: AMIBROKER. In this daily chart of AKAM, uptrend periods are shown using a pale yellow background. Also shown are buy/sell signals generated using 50/70 RSI crossovers during those uptrend periods.

AmiBroker formula language

```
// uptrend defined as moving 20% above
// one year lowest low
// downtrend defined as moving 20% below
// one year highest high

YearlyLow = LLV( C, 252 );
YearlyHigh = HHV( C, 252 );

UpTrendBegin = Close > 1.2 * YearlyLow;
DownTrendBegin = Close < 0.8 * YearlyHigh;

InUpTrend = Flip( UpTrendBegin, DownTrendBegin );

// open new long positions only in uptrend
Buy = InUpTrend AND Cross( 50, RSI( 14 ) );
// sell when uptrend ended or RSI cross
Sell = NOT InUpTrend OR Cross( RSI( 14 ), 70 );

// remove excessive signals
Buy = ExRem( Buy, Sell );
Sell = ExRem( Sell, Buy );

Plot( C, "Price", colorDefault, styleCandle );

Plot( InUpTrend, "", colorLightYellow, styleArea | styleOwnScale | styleNoLabel, 0,
1, 0, -2 );

PlotShapes( Buy * shapeUpArrow, colorGreen, 0, L );
PlotShapes( Sell * shapeDownArrow, colorRed, 0, H );
```

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

BACK TO LIST

NEUROSHELL TRADER: JUNE 2015

The RSI price trends method described by Kevin Luo in his article in this issue, “The RSI & Price Trends,” can be easily implemented with a few of NeuroShell Trader’s 800+ indicators and one indicator from the Turning Points add-on for NeuroShell Trader that implements Luo’s trend condition.

The Turning Point Percent indicator computes the percent change in price between two arbitrary turning points.

```
BUY LONG CONDITIONS:                    RSI > 50
    (both conditions must be true)      TURNING POINT PERCENT  > 10
                                        (value adjusted for example stock)
```

```
SELL LONG CONDITION:                    TURNING POINT PERCENT < 10
```

Users of NeuroShell Trader can go to the Stocks & Commodities section of the NeuroShell Trader free technical support w ebsite to dow nload a copy of this or any previous Traders' Tips.

A sample chart is show n in Figure 5.

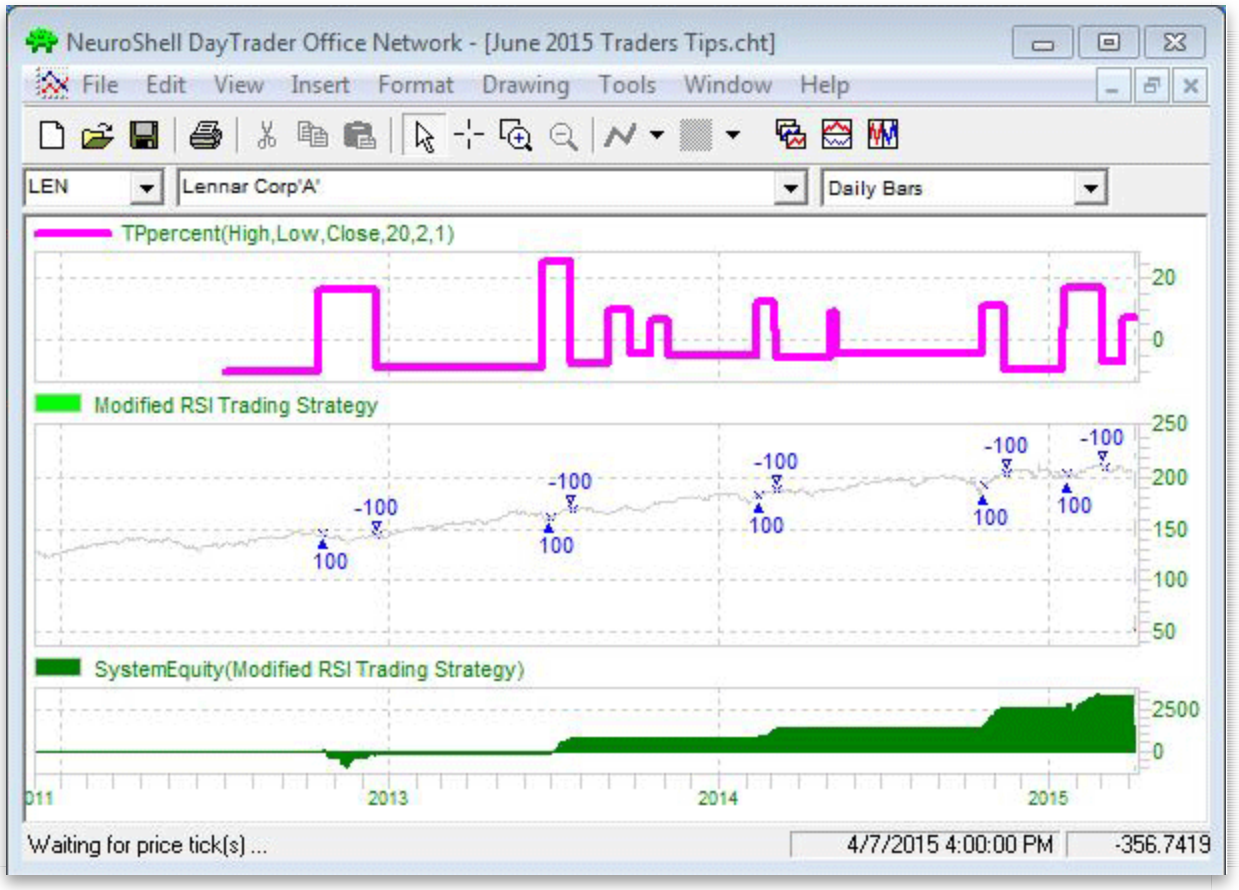


FIGURE 5: NEUROSHELL TRADER. This NeuroShell Trader chart displays the combined RSI and price trends trading system described by Kevin Luo in his article in this issue.

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

BACK TO LIST

NINJATRADER: JUNE 2015

In "The RSI & Price Trends" in this issue, author Kevin Luo presents a strategy based on the relative strength index. We are making available for dow nload a strategy based on the article named "RSIandPriceTrends" at w w w .ninjatrader.com/SC/June2015SC.zip.

Once you have downloaded it, from within the NinjaTrader Control Center window, select the menu **File → Utilities → Import NinjaScript** and select the downloaded file. This file is for NinjaTrader version 7 or greater.

You can review the strategy source code by selecting the menu **Tools → Edit NinjaScript → Strategy** from within the NinjaTrader Control Center window and selecting the "RSIandPriceTrends" file.

A sample chart implementing the strategy is shown in Figure 6.

FIGURE 6: NINJATRADER. This screenshot shows the strategy applied to a daily chart of LEN in NinjaTrader.

—Raymond Deux & Patrick Hodges
NinjaTrader, LLC, www.ninjatrader.com

BACK TO LIST

UPDATA: JUNE 2015

Our Traders' Tips for this month is based on "The RSI & Price Trends" by Kevin Luo. In the article, Luo returns to a technical analysis classic—the relative strength index (RSI)—and overlays some bespoke trading rules

based on observations of price trend and RSI levels. The author proposes that stocks w ith a shorter trend improve the performance of the RSI.

The stock index subset selection can be done in the scan section of the Updata terminal prior to running the system code.

A sample chart is show n in Figure 7.

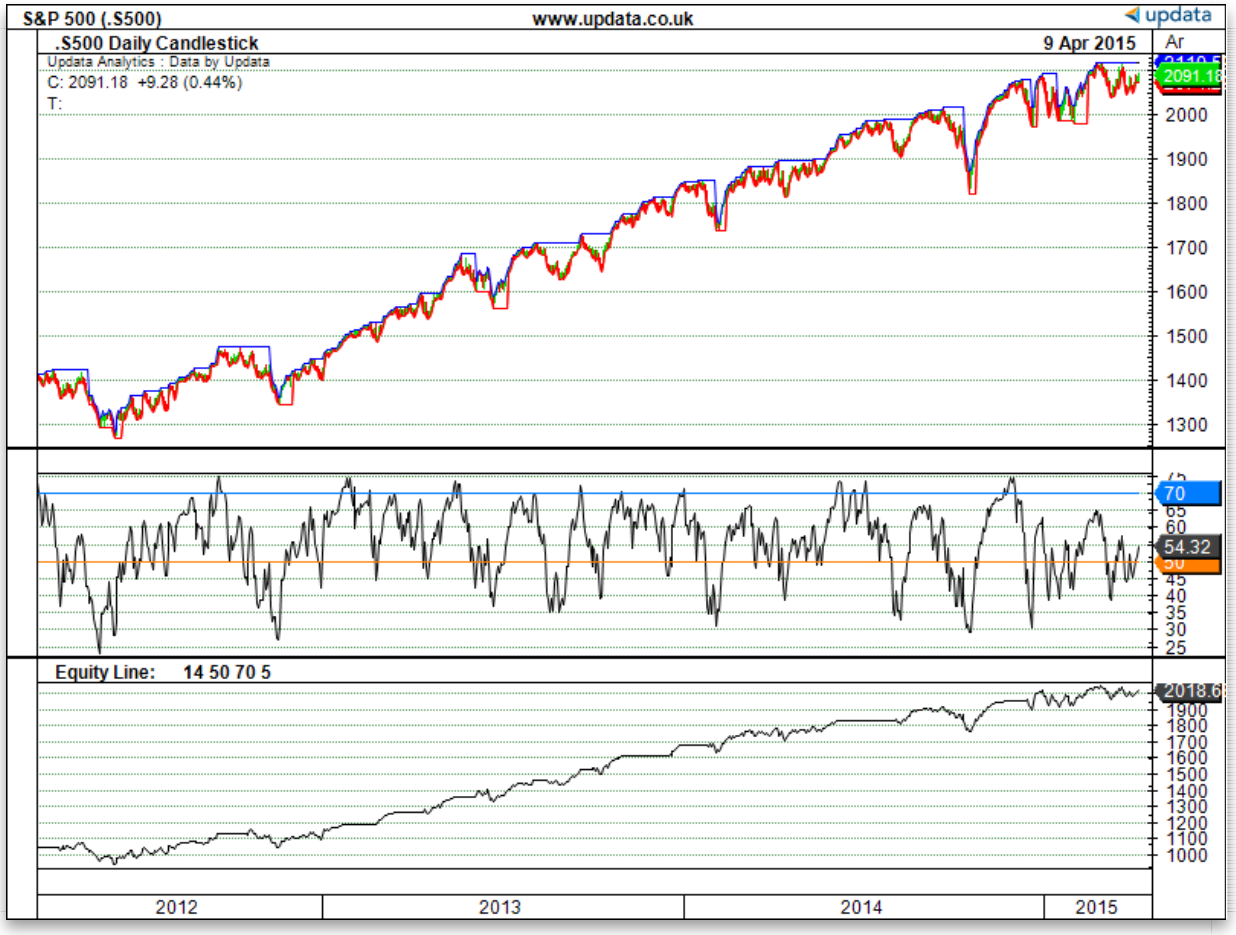


FIGURE 7: UPDATA. This chart shows the trading rules as applied to S&P 500 index data of daily resolution, with 5% trends.

We are making the Updata code for this article available in the Updata library. It may be dow nloaded by clicking the *custom* menu and *system library*. Those w ho cannot access the library due to a firew all may paste the code show n below into the Updata custom editor and save it.

```
PARAMETER "RSI Period" #PERIOD=14
PARAMETER "Buy Level" #BUYLEVEL=50
PARAMETER "Sell Level" #SELLLEVEL=70
PARAMETER "% Move Mode" @PCT=20
DISPLAYSTYLE 5LINES
INDICATORTYPE TOOL
INDICATORTYPE3 CHART
COLOUR RGB(0,0,255)
COLOUR2 RGB(255,0,0)
COLOUR4 RGB(0,0125,255)
```

```
COLOUR5 RGB(255,125,0)
NAME "" ""
@RSI=0
@MODE=1
@LOWPRICE=0
@HIGHPRICE=0
FOR #CURDATE=#PERIOD TO #LASTDATE
   @RSI=RSI(#PERIOD)
   'KEEPS TRACK OF HIGH/LOW POINT IN TREND
   IF @MODE=1 AND HIST(@MODE,1)=1
      @HIGHPRICE=MAX(HIGH,@HIGHPRICE)
      @LOWPRICE=LOW
      IF 100*(@HIGHPRICE-LOW)/@HIGHPRICE>@PCT
         @MODE=-1
      ENDIF
    ELSEIF @MODE=1 AND HIST(@MODE,1)!=1
      @HIGHPRICE=HIGH
   ENDIF
   IF @MODE=-1 AND HIST(@MODE,1)=-1
      @LOWPRICE=MIN(LOW,@LOWPRICE)
       @HIGHPRICE=HIGH
      IF 100*(HIGH-@LOWPRICE)/@LOWPRICE>@PCT
         @MODE=1
      ENDIF
      @LOWPRICE=LOW
   ENDIF
   'ENTRY RULES
   IF @MODE=-1 AND @RSI<#BUYLEVEL
      COVER CLOSE
   ENDIF
   IF @MODE=1 AND @RSI>#SELLLEVEL
      SELL CLOSE
   ENDIF
   IF @MODE=1 AND @RSI<#BUYLEVEL
      BUY CLOSE
   ENDIF
   IF @MODE=-1 AND @RSI>#SELLLEVEL
      SHORT CLOSE
   ENDIF
   @PLOT=@HIGHPRICE
   @PLOT2=@LOWPRICE
   @PLOT3=@RSI
   @PLOT4=#SELLLEVEL
   @PLOT5=#BUYLEVEL
NEXT
```

—Updata support team
support@updata.co.uk, www.updata.co.uk

BACK TO LIST

AIQ: JUNE 2015

The AIQ code I am providing this month is based on Kevin Luo's article in this issue, "The RSI & Price Trends." The code provides a long-only trading system suggested by Luo in his article. It has the following rules:

- Enter trades only during uptrends, that is, when highest high since the lowest low is at least equal to X% from the low of the lowest low in the long-term lookback and when the highest high since the lowest low is at least equal to Y% from the low of the lowest low in the short-term lookback, and
- Enter when the RSI is moving up and crossing above 50
- Close positions when the uptrend reverses, that is, prices move down by more than 20% from the high of the prior uptrend, or
- Close when the RSI crosses down from above 70.

Figure 8 shows the equity curve and key metrics for the system trading the NASDAQ 100 list of stocks from 1/3/2000 to 4/21/2015. Commission and slippage have been subtracted. The system nicely beats the NDX index over the same time period both in return and maximum drawdown.

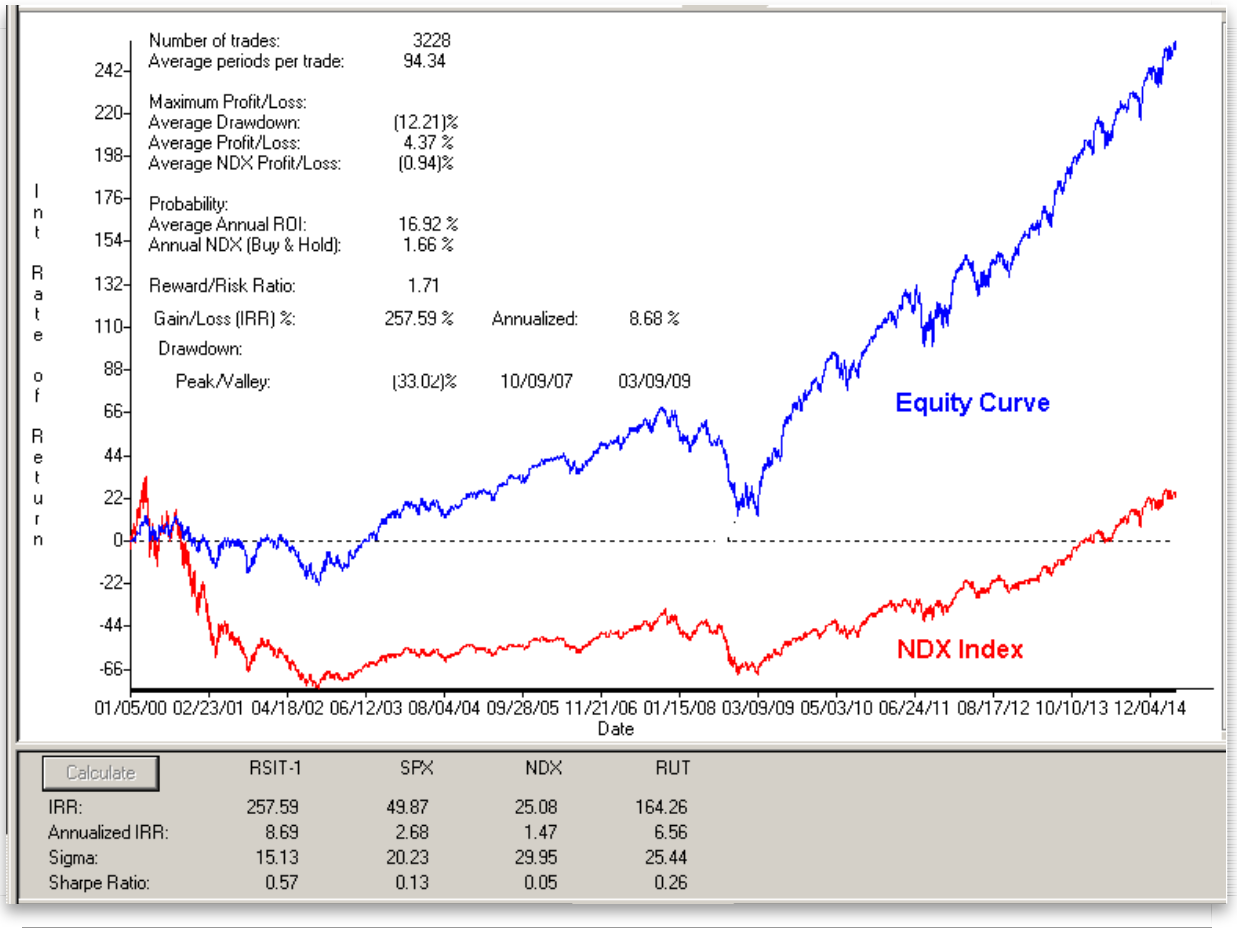


FIGURE 8: AIQ. This shows the equity curve and key metrics for the system trading the NASDAQ 100 list of stocks from 1/3/2000 to 4/21/2015

The code and EDS file can be downloaded from www.TradersEdgeSystems.com/traderstips.htm. The code is as follows:

```
!RSI & PRICE TRENDS

!Author: Kevin Luo, TASC June 2015

!Coded by: Richard Denning 4/21/2015

!www.TradersEdgeSystems.com


!INPUTS

! PARAMETERS:

! ***********Entry Setup Parameters***************

rsiLen  is      14.     ! rsi length

buyLvl  is      50.     ! level for rsi to get a buy setup

LLlen1  is      250.    ! long term lookbakc for lowest low

LLlen2  is      20.     ! short term lookback for lowest low

minPctUp1 is    20.     ! percent up from lowest low in lookback to make an
uptrend

minPctUp2 is    5.     ! percent up from lowest low in lookback to make an uptrend


!************Exit Parameters**********************

exitLvl   is    70.     ! level for rsi to get a buy setup

!**************INDICATOR CODE***************************

!! RSI WILDER

!To convert Wilder Averaging to Exponential Averaging use this formula:

!ExponentialPeriods = 2 * WilderPeriod - 1.


        U       is [close]-val([close],1).

        D       is val([close],1)-[close].


        W1      is rsiLen.

        rsiLen1 is 2 * W1 - 1.

        AvgU    is ExpAvg(iff(U>0,U,0),rsiLen1).

        AvgD    is ExpAvg(iff(D>=0,D,0),rsiLen1).

        rsi     is 100-(100/(1+(AvgU/AvgD))).
```

```
!SYSTEM CODE:

LL1 is loval([low],LLlen1).

LL2 is loval([low],LLlen2).

OSD is offsettodate(month(),day(),year()).

BarsSinceLL1 is scanany([low] = ^LL1,LLlen1) then OSD.

BarsSinceLL2 is scanany([low] = ^LL2,LLlen2) then OSD.

HHsinceLL1 is hival([high],^BarsSinceLL1).

HHsinceLL2 is hival([high],^BarsSinceLL2).

TrendUpLT if (HHsinceLL1 / LL1 - 1)*100 >= minPctUp1.

TrendUpST if (HHsinceLL2 / LL2 - 1)*100 >= minPctUp2.


Buy if TrendUpLT and TrendUpST

        and rsi>buyLvl and valrule(rsi<buyLvl,1).

ExitBuy if (HHsinceLL1 / LL1 - 1)*100 <= minPctUp1

        !or (HHsinceLL2 / LL2 - 1)*100 <= minPctUp2

        or (rsi<exitLvl and valrule(rsi>=exitLvl,1)).

        !or countof(rsi < buyLvl,3,1)=3.

RoC1 is (HHsinceLL1 / LL1 - 1)*100.
```

—Richard Denning
info@TradersEdgeSystems.com
for AIQ Systems

BACK TO LIST

TRADERSSTUDIO: JUNE 2015

The TradersStudio code for Kevin Luo's article "RSI & Price Trends" can be found at:

- www.TradersEdgeSystems.com/traderstips.htm
- www.TradersStudio.com → Traders Resources

The following code files are contained in the download from the websites:

- System: RSITREND - a long only system that uses daily data and buys stocks in the session list that meet the folloing entry and exit criteria:
    - Enter trades only during uptrends, that is, w hen highest high since the low est low is at least equal to X% from low of the low est low in long term lookback and w hen w hen the highest high since the low est low is at least equal to Y% from low of the low est low in short term lookback and
    - Entering w hen the RSI is moving up and crossing above 50.
    - Closing positions w hen uptrend reverses, that is, prices move dow n by more than 20% from the high of prior uptrend or
    - Closing w hen the the RSI crosses dow n from above 70.

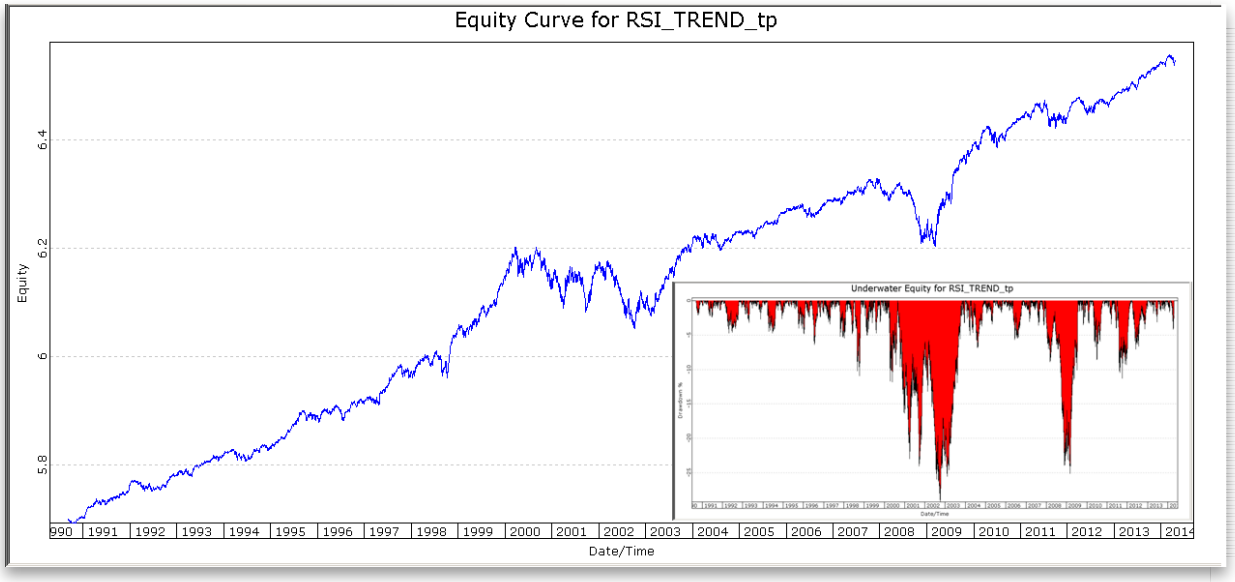


FIGURE 9: LOG EQUITY CURVE AND UNDERWATER EQUITY CURVE FOR THE RSI TREND SYSTEM. Run on the EQUALDOLLAR tradeplan using the NASDAQ 100 list of stocks for the years 1991 through 2014.

Using the NASDAQ 100 list of stocks, I ran the EQUALDOLLAR tradeplan that comes w ith TradersStudio. Figure 9 show s the log chart of the equity curve for the years 1991 through 2014. Also in the same figure is the underw ater equity curve. The system returned about 9% compounded over the period w ith tw o of the largest draw dow ns occurring during the 2000 to 2003 (29%) and the 2007 to 2009 (25%) bear market. Otherw ise the draw dow ns run around 5% – 10%.

The code is also show n here:

```
'RSI & PRICE TRENDS

'Author: Kevin Luo, TASC June 2015

'Coded by: Richard Denning 4/22/2015

'www.TradersEdgeSystems.com
```

```
Sub RSITREND(rsiLen,buyLvl,LLlen1,LLlen2,minPctUp1,minPctUp2,exitLvl)

'rsiLen=14,buyLvl=50,LLlen1=250,LLlen2=20,minPctUp1=20,minPctUp2=5,exitLvl=70

Dim theRSI As BarArray

theRSI = rsi(C,rsiLen,0)

'Dim LL1 As BarArray

'Dim LL2 As BarArray

Dim LL1, LL2, barsSinceLL1, barsSinceLL2,  HHsinceLL1, HHsinceLL2, RoC1, RoC2

Dim trendUpLT As BarArray

Dim trendUpST As BarArray

LL1 = Lowest(L,LLlen1,0)

LL2 = Lowest(L,LLlen2,0)

barsSinceLL1 = MRO(L = LL1,LLlen1,1)

barsSinceLL2 = MRO(L = LL2,LLlen2,1)

HHsinceLL1 = Highest(H,barsSinceLL1,0)

HHsinceLL2 = Highest(H,barsSinceLL2,0)

RoC1 = (HHsinceLL1 / LL1 - 1) * 100

RoC2 = (HHsinceLL2 / LL2 - 1) * 100

trendUpLT = RoC1 >= minPctUp1

trendUpST = RoC2 >= minPctUp2

If trendUpLT And trendUpST And theRSI>buyLvl And theRSI[1]<= buyLvl Then

    Buy("LE",1,0,Market,Day)

End If

If RoC1 < minPctUp1 Then ExitLong("LX_trend","",1,0,Market,Day)

If theRSI<exitLvl And theRSI[1]>=exitLvl Then ExitLong("LX_rsi","",1,0,Market,Day)

End Sub
```

Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

THINKORSWIM: JUNE 2015

In his article titled “The RSI & Price Trends”, Kevin Luo attempts to build a w inning strategy around an old technical analysis tool RSI. We have built his RSITrend Strategy using our proprietary scripting language, thinkscript. We have made the loading process extremely easy, simply click on the link here http://tos.mx/PKDtbn and choose Backtest in thinkorsw im, then choose to rename your Strategy to RSITrend. You can adjust the parameters of these w ithin the *Edit Studies* w indow to fine tune your variables.

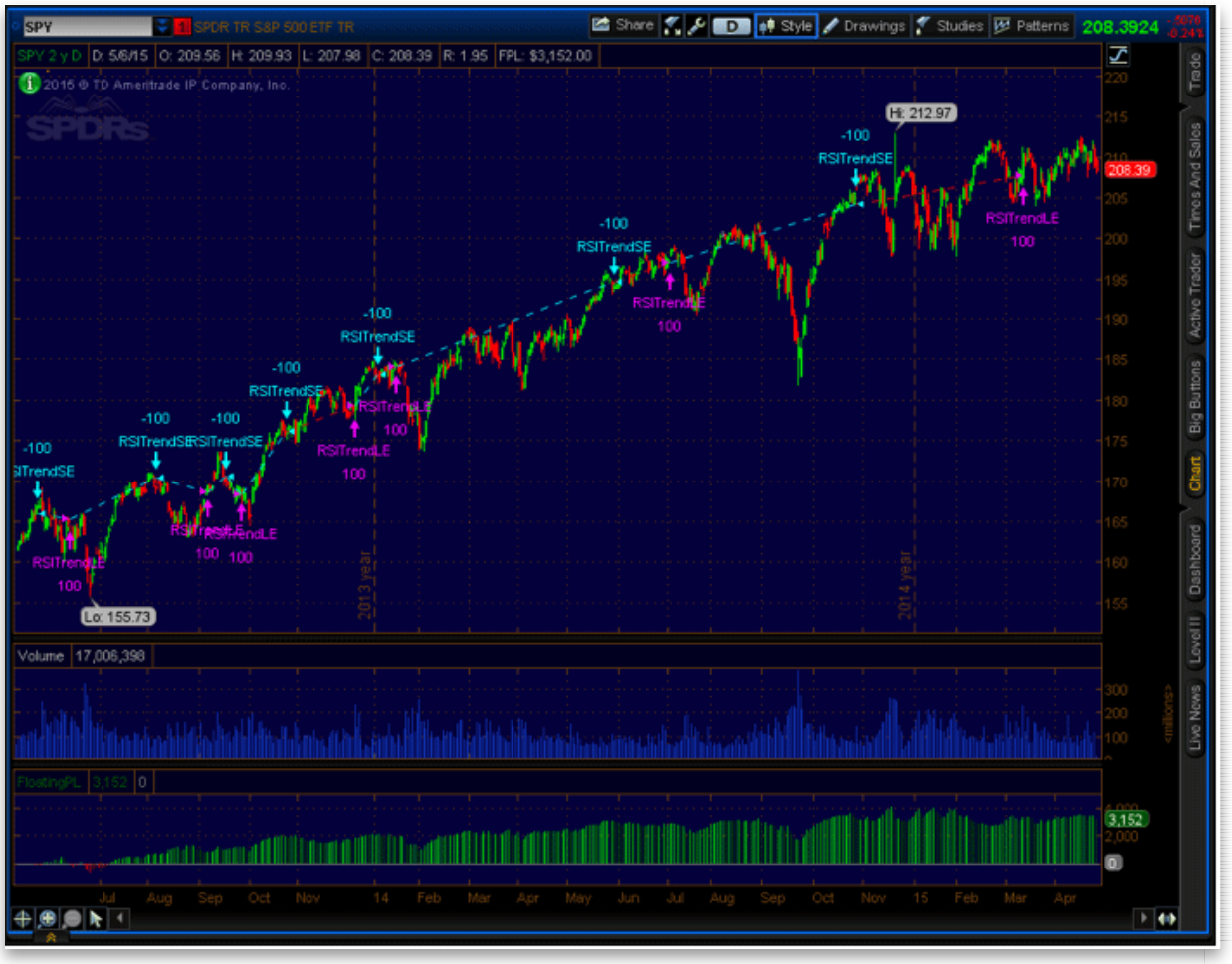


FIGURE 10: THINKORSWIM.

In the example above (Figure 10) w e see entry and exit points based on the logic w hich the article defined. This is for symbol SPY, and you can see the green historgram below the volume displays the profit based on this strategy. For a detailed description see the article in *Technical Analysis of* STOCKS & COMMODITIES Magazine.

—thinkorswim
A division of TD Ameritrade, Inc.
www.thinkorswim.com

BACK TO LIST

Originally published in the June 2015 issue of
Technical Analysis of STOCKS & COMMODITIES magazine.

July 2015

For this month's Traders' Tips, the focus is mainly Vitali Apirine's article that appeared in the April 2015 issue of STOCKS & COMMODITIES, "The Slow Relative Strength Index." Other Traders' Tips are based on different S&C articles as indicated. Here, we present the July 2015 Traders' Tips code with possible implementations in various software.

The Traders' Tips section is provided to help the reader implement a selected technique from an article in this issue or another recent issue. The entries here are contributed by software developers or programmers for software that is capable of customization.

TRADESTATION: JULY 2015

In "The Slow Relative Strength Index," which appeared in the April 2015 issue of STOCKS & COMMODITIES, author Vitali Apirine describes a modified version of the well-known relative strength index (RSI). Apirine describes a method of smoothing the price used for the RSI calculation by using an exponential moving average of price.

For convenience, we are providing the code for a TradeStation function (SRSI) based on the author's ideas. This function can be used in other analysis techniques or strategies. We have also provided a demonstration indicator using the SRSI function. The EasyLanguage code is as follows:

```
Function: SRSI

// TASC JUL 2015
// SRSI

inputs:
        iPrice( numericseries ),
        Length( numericsimple ),{ this input assumed to be a constant >= 1 }
        SMoothingLength( numericsimple ) ;
```

```
variables:
        NetChgAvg( 0 ),
        TotChgAvg( 0 ),
        Change( 0 ),
        SF( 1 / Length ), { smoothing factor }
        ChgRatio( 0 ),
        Price( 0 ) ;

Price = XAverage( iPrice, SMoothingLength ) ;

if CurrentBar = 1 then
        begin
        NetChgAvg = ( Price - Price[Length] ) / Length ;
        TotChgAvg = Average( AbsValue( Price - Price[1] ), Length ) ;
        end
else
        begin
        Change = Price - Price[1] ;
        NetChgAvg = NetChgAvg[1] + SF * ( Change - NetChgAvg[1] ) ;
        TotChgAvg = TotChgAvg[1] + SF * ( AbsValue( Change ) - TotChgAvg[1] ) ;
        end ;

if TotChgAvg <> 0 then
        ChgRatio = NetChgAvg / TotChgAvg
else
        ChgRatio = 0 ;

SRSI = 50 * ( ChgRatio + 1 ) ;
```

Indicator: SRSI

```
// TASC JUL 2015
// SRSI

inputs:
        iPrice( numericseries ),
        Length( numericsimple ),{ this input assumed to be a constant >= 1 }
        SMoothingLength( numericsimple ) ;

variables:
        NetChgAvg( 0 ),
        TotChgAvg( 0 ),
        Change( 0 ),
        SF( 1 / Length ), { smoothing factor }
        ChgRatio( 0 ),
        Price( 0 ) ;

Price = XAverage( iPrice, SMoothingLength ) ;

if CurrentBar = 1 then
        begin
        NetChgAvg = ( Price - Price[Length] ) / Length ;
        TotChgAvg = Average( AbsValue( Price - Price[1] ), Length ) ;
        end
else
        begin
        Change = Price - Price[1] ;
        NetChgAvg = NetChgAvg[1] + SF * ( Change - NetChgAvg[1] ) ;
        TotChgAvg = TotChgAvg[1] + SF * ( AbsValue( Change ) - TotChgAvg[1] ) ;
```

```
        end ;

if TotChgAvg <> 0 then
        ChgRatio = NetChgAvg / TotChgAvg
else
        ChgRatio = 0 ;

SRSI = 50 * ( ChgRatio + 1 ) ;
```

To dow nload the EasyLanguage code, please visit our TradeStation and EasyLanguage support forum. The code from this article can be found at http://w w w .tradestation.com/TASC-2015. The ELD filename is "TASC_JUL2015.ELD."

For more information about EasyLanguage in general, please see http://w w w .tradestation.com/EL-FAQ.

A sample chart is show n in Figure 1.

FIGURE 1: TRADESTATION. This shows the SRSI indicator

along with the standard RSI indicator for comparison, applied to the daily S&P 500 index.

This article is for informational purposes. No type of trading or investment recommendation, advice, or strategy is being made, given, or in any manner provided by TradeStation Securities or its affiliates.

—Doug McCrary
TradeStation Securities, Inc.
www.TradeStation.com

BACK TO LIST

eSIGNAL: JULY 2015

For this month's Traders' Tip, we're providing the study SRSI.efs based on Vitali Apirine's article in the April 2015 issue of STOCKS & COMMODITIES, "The Slow Relative Strength Index." In it, Apirine applies an exponential moving average to smooth J. Welles Wilder's classic relative strength index (RSI).

The study requires eSignal 12.1 or higher and contains formula parameters that may be configured through the *edit chart* window (right-click on the chart and select "edit chart"). A sample chart implementing the study is shown in Figure 2.

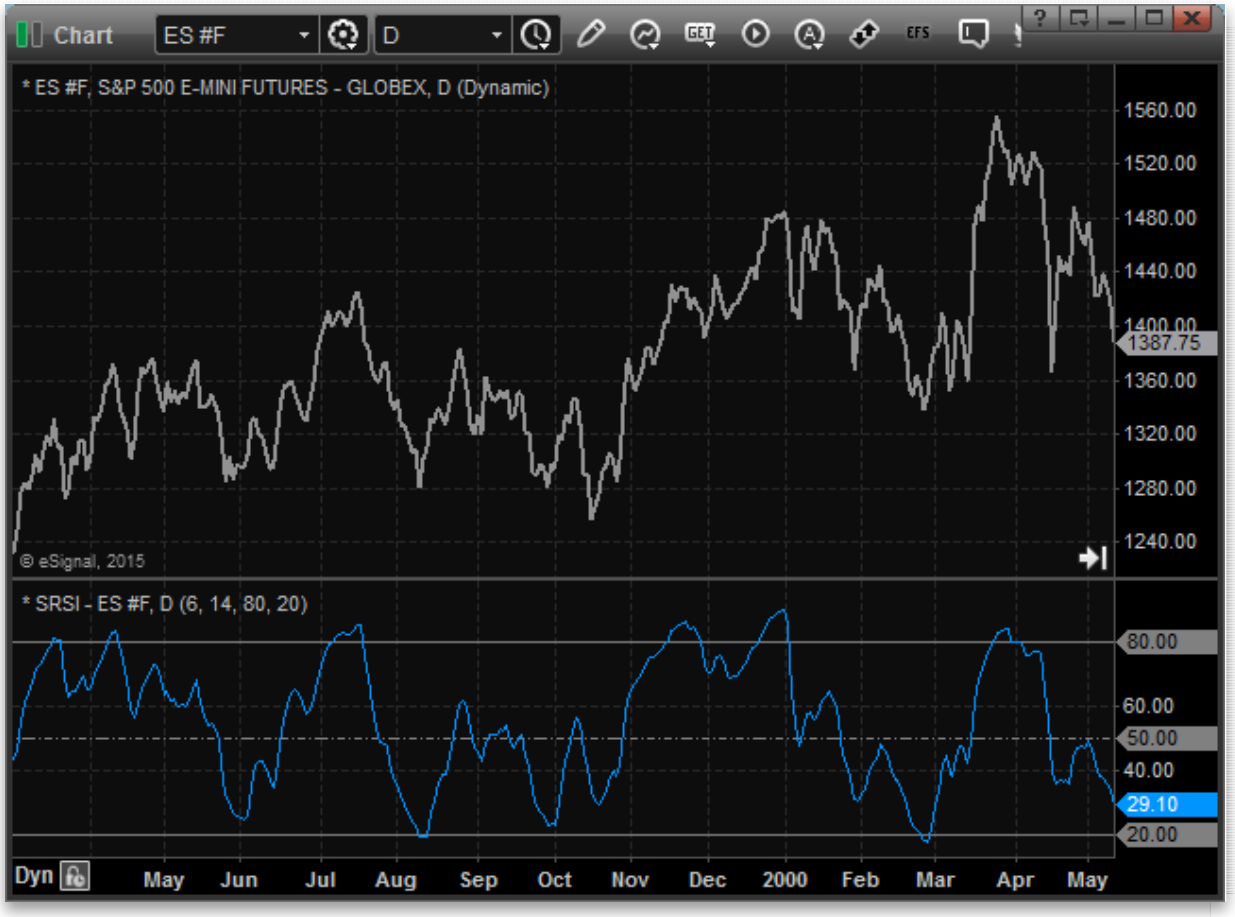


FIGURE 2: eSIGNAL. Here is an example of the study plotted on a chart of the S&P 500 emini (ES).

To dow nload this study or a copy of the EFS code, please visit our EFS library. To discuss this formula, please visit the EFS studies forum.

The code is also show n here:

```
/**********************************
Provided By:
    Interactive Data Corporation (Copyright B© 2015)
    All rights reserved. This sample eSignal Formula Script (EFS)
    is for educational purposes only. Interactive Data Corporation
    reserves the right to modify and overwrite this EFS file with
    each new release.

Description:
    The Slow Relative Strength Index by Vitali Apirine

Formula Parameters:                     Default:
Length EMA                              6
Length Average Differences              14
Upper Bound                             80
Lower Bound                             20

Version:            1.00  05/05/2015

Notes:
    The related article is copyrighted material. If you are not a subscriber
    of Stocks & Commodities, please visit www.traders.com.

**********************************/

var fpArray = new Array();

function preMain(){

    setStudyTitle("SRSI");

    setCursorLabelName("Upper Bound", 0);
    setCursorLabelName("Lower Bound", 1);
    setCursorLabelName("Center Line", 2);
    setCursorLabelName("SRSI", 3);

    setDefaultBarFgColor(Color.grey, 0);
    setDefaultBarFgColor(Color.grey, 1);
    setDefaultBarFgColor(Color.grey, 2);

    setShowCursorLabel(false, 0);
    setShowCursorLabel(false, 1);
    setShowCursorLabel(false, 2);
    setShowCursorLabel(true, 3);

    setDefaultBarStyle(PS_SOLID, 0);
    setDefaultBarStyle(PS_SOLID, 1);
    setDefaultBarStyle(PS_DASHDOT, 2);
    setDefaultBarStyle(PS_SOLID, 3);
```

```
    var x = 0;

    fpArray[x] = new FunctionParameter("fpLengthEMA", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("Length EMA");
        setLowerLimit(1);
        setDefault(6);
    };

    fpArray[x] = new FunctionParameter("fpLengthAvgDiff", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("Length Average Differences");
        setLowerLimit(1);
        setDefault(14);
    };

    fpArray[x] = new FunctionParameter("fpSRSIHighBorder", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("Upper Bound");
        setLowerLimit(0);
        setUpperLimit(100);
        setDefault(80);
    };

    fpArray[x] = new FunctionParameter("fpSRSILowBorder", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("Lower Bound");
        setLowerLimit(0);
        setUpperLimit(100);
        setDefault(20);
    };
}

var bInit = false;
var bVersion = null;

var xDifferences = null;
var xPositiveDiff = null;
var xNegativeDiff = null;
var xPositiveDiffAvg = null;
var xNegativeDiffAvg = null;

function main(fpLengthEMA, fpLengthAvgDiff, fpSRSIHighBorder, fpSRSILowBorder){

    if (bVersion == null) bVersion = verify();
    if (bVersion == false) return;

    if (!bInit){

        xDifferences = efsInternal('Calc_Differences', fpLengthEMA);
        xPositiveDiff = getSeries(xDifferences, 0);
        xNegativeDiff = getSeries(xDifferences, 1);

        xPositiveDiffAvg = smma(fpLengthAvgDiff, xPositiveDiff);
        xNegativeDiffAvg = smma(fpLengthAvgDiff, xNegativeDiff);

        bInit = true;
    };
```

```
    var nPositiveDiffAvg = xPositiveDiffAvg.getValue(0);
    var nNegativeDiffAvg = xNegativeDiffAvg.getValue(0);

    if (nPositiveDiffAvg == null || nNegativeDiffAvg == null)
        return;

    var nSRSI = (nNegativeDiffAvg == 0) ? 100 : 100 - (100 / (1 + (nPositiveDiffAvg /
nNegativeDiffAvg)));

    return [fpSRSIHighBorder, fpSRSILowBorder, 50, nSRSI];
}

var xClose = null;
var xEMA = null;

function Calc_Differences(nLength){

    if (getBarState() == BARSTATE_ALLBARS){
        xClose = close();
        xEMA = ema(nLength);
    }

    var nClose = xClose.getValue(0);
    var nEMA = xEMA.getValue(0);

    if (nClose == null || nEMA == null)
        return;

    var nPositiveDiff = nClose > nEMA ? nClose - nEMA : 0;
    var nNegativeDiff = nClose < nEMA ? nEMA - nClose : 0;

    return [nPositiveDiff, nNegativeDiff];
}

function verify(){
    var b = false;
    if (getBuildNumber() < 3742){

        drawTextAbsolute(5, 35, "This study requires version 12.1 or later.",
            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,
            null, 13, "error");
        drawTextAbsolute(5, 20, "Click HERE to upgrade.@URL=http://www.esignal.com
/download/default.asp",
            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,
            null, 13, "upgrade");
        return b;
    }
    else
        b = true;

    return b;
}
```

—Eric Lippert
eSignal, an Interactive Data company
800 779-6555, www.eSignal.com

BACK TO LIST

THINKORSWIM: JULY 2015

In “The Slow Relative Strength Index,” w hich appeared in the April 2015 issue of STOCKS & COMMODITIES, author Vitali Apirine provided a new look to a topical, old technical analysis tool, the relative strength index (RSI). He uses the analogy “slow and steady w ins the race” to explain how to build a new RSI, w hich has a slow ing or smoothing mechanism.

We have recreated his slow relative strength index (SRSI) in a Slow RSI study using our proprietary scripting language, thinkscript. We have made the loading process extremely easy. Simply click on the link http://tos.mx/Wu94qa and choose *save script to thinkorswim*, then choose to rename your study to “Slow RSI.” You can adjust the parameters of these w ithin the *edit studies* w indow to fine-tune your variables.

In the example show n in Figure 3, you see the slow RSI plotted on a chart of the S&P 500 index over the past year. It is easy to see that the SRSI is much smoother than the traditional RSI. For more details about the SRSI, see Apirine’s article in the April 2015 issue, available at Traders.com.

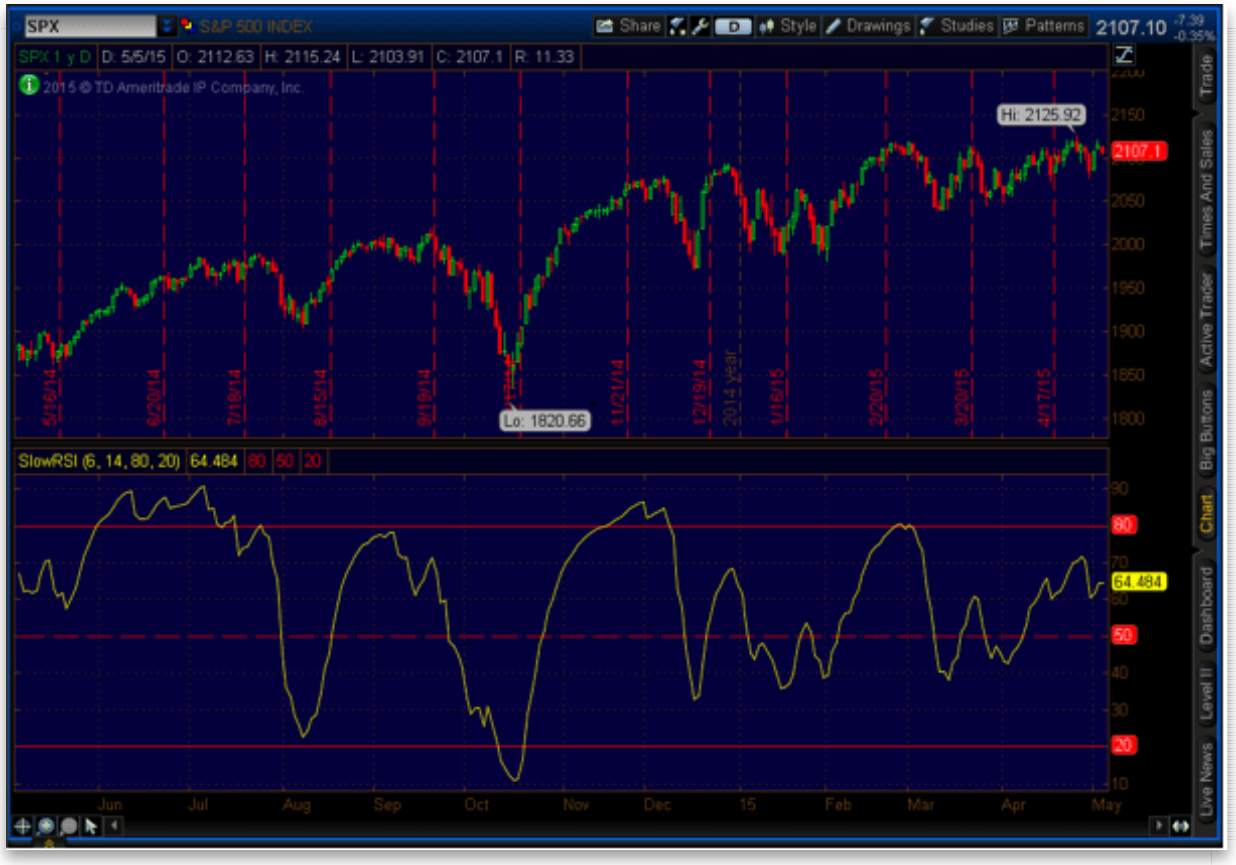


FIGURE 3: THINKORSWIM. Here is an example of the slow RSI (SRSI) plotted on a chart of the S&P 500 index over the past year. It is easy to see that the SRSI is much smoother than the traditional RSI.

THINKORSWIM: JUNE 2015

In “RSI & Price Trends,” which appeared in the June 2015 issue of STOCKS & COMMODITIES, author Kevin Luo attempts to build a winning strategy around an old technical analysis tool, the RSI. We have recreated his RSITrend strategy using our proprietary scripting language, *thinkscript*.

We have made the loading process extremely easy. Simply click on the link http://tos.mx/PKDtbn and choose *backtest in thinkorswim*, then choose to rename your strategy to “RSITrend.” You can adjust the parameters within the *edit studies* window to fine-tune your variables.

In the example chart shown in Figure 4, we see entry & exit points based on the logic, which the article defined. This example is for symbol SPY. The green histogram below the volume displays the profit based on this strategy. For a more detailed description of the strategy, see Luo’s article in the June 2015 issue of S&C or at the Traders.com website.

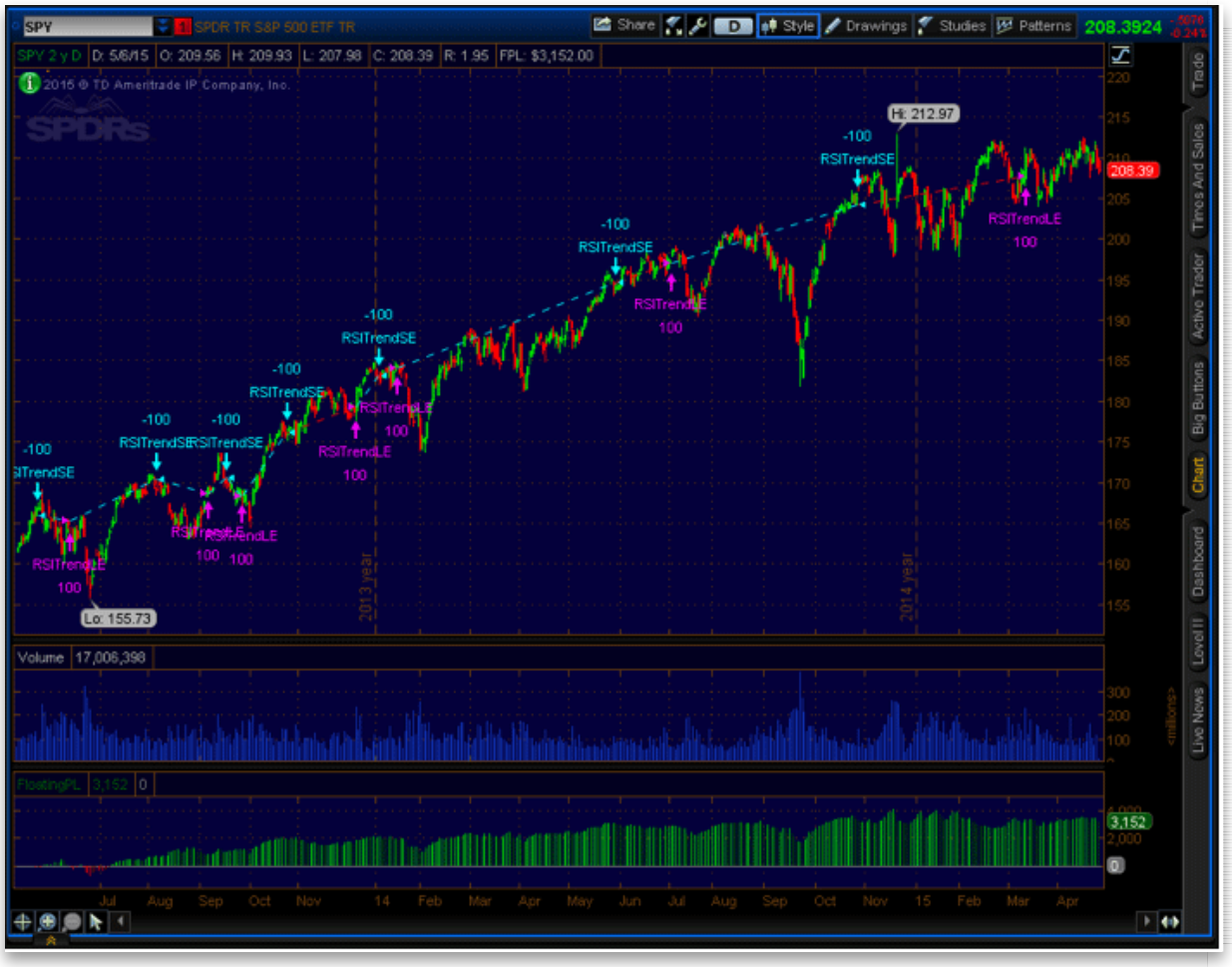


FIGURE 4: THINKORSWIM. Here are sample entry & exit points on the SPY based on the logic behind Luo's RSITrend strategy as described in his June 2015 article. The green histogram below the volume displays the profit based on this strategy.

—thinkorswim
A division of TD Ameritrade, Inc.
www.thinkorswim.com

BACK TO LIST

WEALTH-LAB: JULY 2015

The slow relative strength index (SRSI) presented by Vitali Apirine in his April 2015 STOCKS & COMMODITIES article, "The Slow Relative Strength Index," is a momentum price oscillator similar to J. Welles Wilder's classic relative strength index (RSI) in its application and interpretation. Oscillating between zero and 100, it becomes overbought after reaching 80 and oversold after dropping below 20. Signals can also be generated by looking for centerline crossovers and divergences. The latter trait is what our example trading system will zero in on.

There are several approaches that help identify divergences between price and oscillator. The one we're going to use is straightforward, detecting a divergence when the SRSI indicator fails to confirm a price extreme, that is, the highest high of 20 days for short trades or the 20-day lowest low for long trades. If we were to rely on finding retracements from a recent peak or trough, divergence detection would introduce a little delay compared to this technique.

As the author suggests, bullish/bearish divergences generated by SRSI are not as effective during strong trends. To avoid fading an established trend, the system is used in conjunction with the average directional movement index (ADX) as trend-confirmation tool. If the ADX is below its threshold for a trending market, the system would enter the trade (Figure 5).

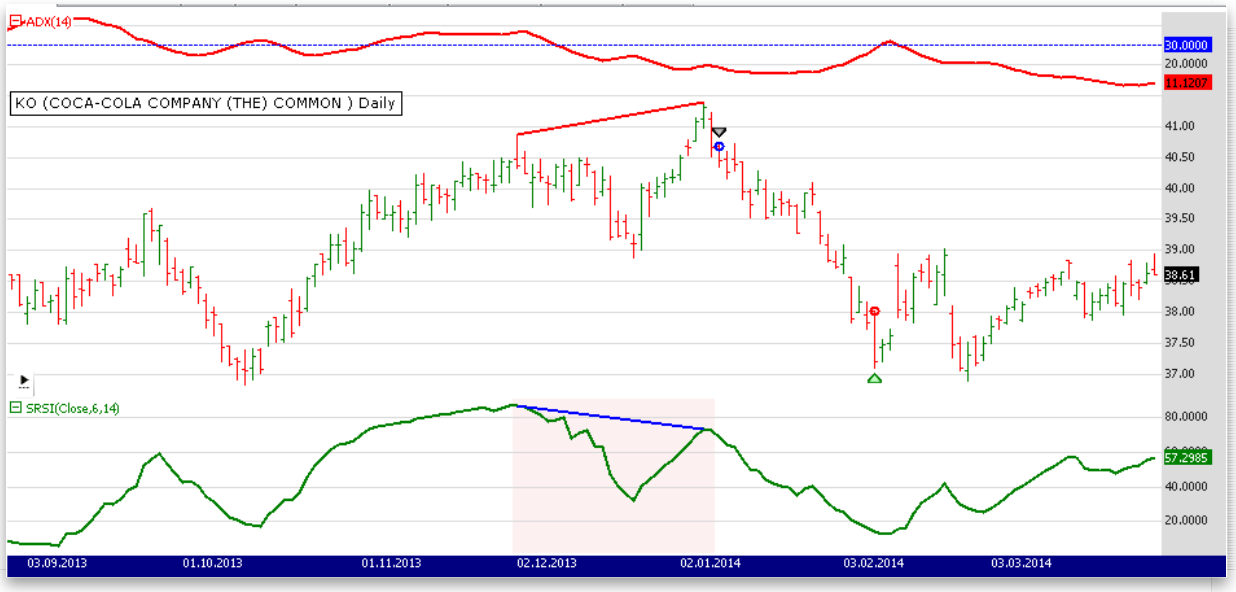


FIGURE 5: WEALTH-LAB. A bearish divergence that formed in January 2014 between the SRSI and price triggered a short trade in KO (Coca-Cola).

After updating the TASCIndicators library to v2015.06 or later in Wealth-Lab, the SRSI indicator can be found under the "TASC Magazine Indicators" group. You can plot SRSI on a chart or use it as an entry or exit condition in a rule-based strategy without having to program any code yourself.

WEALTH-LAB CODE:

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Drawing;
using WealthLab;
using WealthLab.Indicators;
using TASCIndicators;

namespace WealthLab.Strategies
{
        /*
                SRSI divergence:
                Price sets a lowest low but the indicator fails to confirm the new
low and turns up
        */

        public class SRSI_Divergence : WealthScript
        {
                private StrategyParameter paramHighest;
                private StrategyParameter paramPeriod;
                private StrategyParameter paramPeriodWMA;
                private StrategyParameter paramThresholdForTrend;
                private StrategyParameter paramExitDays;

                public SRSI_Divergence()
                {
                        paramPeriod = CreateParameter("SRSI period", 6, 1, 100, 1);
                        paramPeriodWMA = CreateParameter("WilderMA period", 14, 1,
100, 1);
                        paramThresholdForTrend = CreateParameter("ADX Threshold", 30,
10, 50, 10);
                        paramHighest = CreateParameter("Highest high of", 20, 5, 50,
1);
                        paramExitDays = CreateParameter("Exit after", 20, 1, 50, 1);
                }

                protected override void Execute()
                {
                        bool peak = false; int peakBar = -1;
                        int high = paramHighest.ValueInt;
                        bool trough = false; int troughBar = -1;
                        int low = paramHighest.ValueInt;
                        int period = paramPeriod.ValueInt;
                        int periodWMA = paramPeriodWMA.ValueInt;
                        int days = paramExitDays.ValueInt;
                        int thresholdForTrend = paramThresholdForTrend.ValueInt;
```

```
                        ADX adx = ADX.Series(Bars,periodWMA);
                        SRSI srsi = SRSI.Series( Close, period, periodWMA );
                        Lowest indicatorLowest = Lowest.Series( srsi, low );
                        Lowest hLow = Lowest.Series( Low, low );

                        HideVolume(); LineStyle solid = LineStyle.Solid;
                        ChartPane srsiPane = CreatePane( 50, false, true );
                        PlotSeries( srsiPane, srsi, Color.Green, solid, 2 );
                        ChartPane adxPane = CreatePane( 25, true, true );
                        PlotSeries( adxPane, adx, Color.Red, solid, 2 );

DrawHorzLine(adxPane,thresholdForTrend,Color.Blue,LineStyle.Dashed,1);

                        for(int bar = GetTradingLoopStartBar(period); bar <
Bars.Count; bar++)
                        {
                                if (!IsLastPositionActive)
                                {
                                        /* 1st peak: both price and indicator */

                                        if( peak == false )
                                        {
                                                if( ( High[bar-1] == Highest.Series(
High, high )[bar-1] )
                                                        & ( srsi[bar-1] ==
Highest.Series( srsi, high )[bar-1] )
                                                        & TurnDown( bar, High ) &
TurnDown( bar, srsi ) )
                                                {
                                                        peak = true; peakBar = bar-1;
                                                }
                                        }

                                        if( peak == true )
                                        {
                                                if( ( High[bar] != Highest.Series(
High, high )[bar] )
                                                        & ( srsi[bar] ==
Highest.Series( srsi, high )[bar] ) )
                                                        peak = false;
                        }

                                        /* 2nd peak: price high not confirmed by the
indicator */

                                        if( peak == true )
                                        {
                                                if( ( High[bar-1] == Highest.Series(
High, high )[bar-1] )
                                                        & ( High[bar-1] >=
High[peakBar] )
                                                        & ( srsi[bar-1] !=
Highest.Series( srsi, high )[bar-1] )
                                                        & ( srsi[bar-1] <
srsi[peakBar] ) &
                                                        TurnDown( bar, High ) &
TurnDown( bar, srsi ) )
                                                {
                                                        peak = false;
```

```
                                                        /* Don't fade a strong trend
*/

                                                        if( adx[bar] <
thresholdForTrend )
                                                                ShortAtMarket( bar+1
);

                                                        /* Highlight divergence */

                                                        for (int b = peakBar; b <=
bar; b++)

SetPaneBackgroundColor( srsiPane, b, Color.FromArgb( 30, Color.LightCoral ) );

                                                        DrawLine( PricePane, peakBar,
High[peakBar], bar-1, High[bar-1], Color.Red, solid, 2 );
                                                        DrawLine( srsiPane, peakBar,
srsi[peakBar], bar-1, srsi[bar-1], Color.Blue, solid, 2 );
                                                }
                                        }

                                        /* 1st trough: both price and indicator */

                                        if( trough == false )
                                        {
                                                if( ( Low[bar-1] == Lowest.Series(
Low, low )[bar-1] )
                                                        & ( srsi[bar-1] ==
Lowest.Series( srsi, low )[bar-1] )
                                                        & TurnUp( bar, Low ) &
TurnUp( bar, srsi ) )
                                                {
                                                        trough = true; troughBar =
bar-1;
                                                }
                                        }

                                        if( trough == true )
                                        {
                                                if( ( Low[bar] != Lowest.Series( Low,
low )[bar] )
                                                        & ( srsi[bar] ==
Lowest.Series( srsi, low )[bar] ) )
                                                        trough = false;
                        }

                                        /* 2nd trough: price low not confirmed by the
indicator */

                                        if( trough == true )
                                        {
                                                if( ( Low[bar-1] == Lowest.Series(
Low, low )[bar-1] )
                                                        & ( Low[bar-1] <=
Low[troughBar] )
                                                        & ( srsi[bar-1] !=
Lowest.Series( srsi, low )[bar-1] )
                                                        & ( srsi[bar-1] >
srsi[troughBar] ) &
```

```
                                                                TurnUp( bar, Low ) & TurnUp(
bar, srsi ) )
                                                        {
                                                                trough = false;

                                /* Don't fade a strong trend */

                                                                if( adx[bar] <
thresholdForTrend )
                                                                        BuyAtMarket( bar+1 );

                                                                /* Highlight divergence */

                                                                for (int b = troughBar; b <=
bar; b++)

SetPaneBackgroundColor( srsiPane, b,

Color.FromArgb( 30, Color.LightGreen ) );

                                                                DrawLine( PricePane,
troughBar, Low[troughBar],
                                                                        bar-1, Low[bar-1],
Color.Blue, solid, 2 );
                                                                DrawLine( srsiPane,
troughBar, srsi[troughBar],
                                                                        bar-1, srsi[bar-1],
Color.Red, solid, 2 );
                                                        }
                                                }
                                        } else
                                        {
                                                /* Exit after N days */

                                                Position p = LastPosition;
                                                if ( bar+1 - p.EntryBar >= days )
                                                        ExitAtMarket( bar+1, p, "Timed" );
                                        }
                                }
                        }
                }
}
```

—Eugene, Wealth-Lab team
MS123, LLC
www.wealth-lab.com

BACK TO LIST

AMIBROKER: JULY 2015

In “The Slow Relative Strength Index” in the April 2015 issue of S&C, author Vitali Apirine presents a “slow” version of the classic RSI indicator. A ready-to-use AmiBroker formula is shown here. To use it, enter the code in the formula editor and press *apply indicator*. You can adjust the exponential average period and the RSI smoothing period using the *parameters* window.

A sample chart is show n in Figure 6.

FIGURE 6: AMIBROKER. Here is the daily slow RSI (6,14) of the S&P 500 (upper pane) with a daily price chart of the S&P 500 during 2011 with breakouts and divergences.

AmiBroker code:

```
// Slow RSI

periods = Param( "Period", 6, 1, 100 );
smooth =  Param( "Smoothing", 14, 1, 100 );

// use current symbol close
price = Close;

// but can use other security data too
// price =  Foreign( "^SPX", "C" );

R1 = EMA( price, periods );
R2 = IIf( price > r1, price - R1, 0 );
R3 = iIf( price < r1, r1 - price, 0 );
R4 = Wilders( R2, smooth );
```

```
R5 = Wilders( R3, smooth );
RR = IIf( R5 == 0, 100, 100 - ( 100 / ( 1 + ( R4 / R5 ) ) ) );

Plot( RR, "SlowRSI" + _PARAM_VALUES(), colorRed );
```

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

BACK TO LIST

AIQ: JULY 2015

The AIQ code based on Vitali Apirine's article in the April 2015 issue of STOCKS & COMMODITIES, "The Slow Relative Strength Index," is shown here. This code for the slow RSI (SRSI) is for use as an indicator. A sample chart illustrating the SRSI is shown in Figure 7.

FIGURE 7: AIQ. This example chart shows the slow RSI (6,14) compared to the classic RSI (14).

```
!THE SLOW RELATIVE STRENGTH INDEX
!Author: Vitali Aprine, TASC April 2015
!Coded by: Richard Denning 5/3/2015
!www.TradersEdgeSystems.com

!INPUTS FOR INDICATOR:
```

```
emaLen is 6.
wilderLen is 14.

!INDICATOR FORMULAS:
ema is expavg([close],emaLen).
pDif is iff([close] - ema > 0,[close] - ema,0).
nDif is iff([close] - ema < 0,ema - [close],0).

rsiLen  is 2 * wilderLen - 1.
AvgU    is expavg(pDif,rsiLen).
AvgD    is expavg(nDif,rsiLen).
srsi    is 100-(100/(1+(AvgU/AvgD))). !PLOT
```

The code and EDS file can be downloaded from www.TradersEdgeSystems.com/traderstips.htm.

—Richard Denning
info@TradersEdgeSystems.com
for AIQ Systems

BACK TO LIST

TRADERSSTUDIO: JULY 2015

The TradersStudio code based on Vitali Apirine's article that appeared the April 2015 issue of STOCKS & COMMODITIES, "The Slow Relative Strength Index," is provided at the following two websites:

- www.TradersEdgeSystems.com/traderstips.htm
- www.TradersStudio.com → Traders Resources

The following code files are provided in the download:

- **Function: SlowRSI**—Computes the slow RSI (SRSI) values
- **Indicator plot: SlowRSI_IND**—Plots the SRSI indicator on a chart.

The code is also shown here:

```
'THE SLOW RELATIVE STRENGTH INDEX
'Author: Vitali Aprine, TASC April 2015
'Coded by: Richard Denning 5/3/2015
'www.TradersEdgeSystems.com

function SlowRSI(emaLen, wilderLen)
'emaLen = 6, wilderLen = 14

'INDICATOR FORMULAS:
Dim ema,pDif,nDif,rsiLen,AvgU,AvgD
Dim sRSI As BarArray
ema = xaverage(C,emaLen)
pDif = IIF(C - ema > 0,C - ema,0)
nDif = IIF(C - ema < 0,ema - C,0)
```

```
rsiLen  = 2 * wilderLen - 1
AvgU    = xaverage(pDif,rsiLen)
AvgD    = xaverage(nDif,rsiLen)
if avgd <> 0 then sRSI    = 100-(100/(1+(AvgU/AvgD)))
SlowRSI = sRSI
End Function
'----------------------------------------
'INDICATOR PLOT CODE:
sub SlowRSI_IND(emaLen, rsiLen)
plot1(slowRSI(emaLen, rsiLen))
plot2(70)
plot3(30)
End Sub
'----------------------------------------
```

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

TRADERSSTUDIO: JUNE 2015

The TradersStudio code based on Kevin Luo's article in the June 2015 issue of STOCKS & COMMODITIES, "RSI & Price Trends," is provided at the following websites:

- www.TradersEdgeSystems.com/traderstips.htm
- www.TradersStudio.com → Traders Resources

The following code file is provided in the download:

- **System: RSITREND**—A long-only system that uses daily data and buys stocks in the session list that meet the following entry & exit criteria:
    - Enter trades only during uptrends, that is, when highest high since the lowest low is at least equal to X% from low of the lowest low in long term lookback and when when the highest high since the lowest low is at least equal to Y% from low of the lowest low in short term lookback, and
    - Entering when the RSI is moving up and crossing above 50
    - Closing positions when uptrend reverses, that is, prices move down by more than 20% from the high of prior uptrend, or
    - Closing when the the RSI crosses down from above 70.

Using the NASDAQ 100 list of stocks, I ran the EQUALDOLLAR tradeplan that comes with the product. Figure 8 shows the log chart of the equity curve for the years 1991 through 2014. The associated underwater equity curve is shown in the same figure. The system returned about 9% compounded over the period with two of the largest drawdowns occurring during the 2000–2003 (29%) and the 2007–2009 (25%) bear market. Otherwise, the drawdowns run around 5–10%.

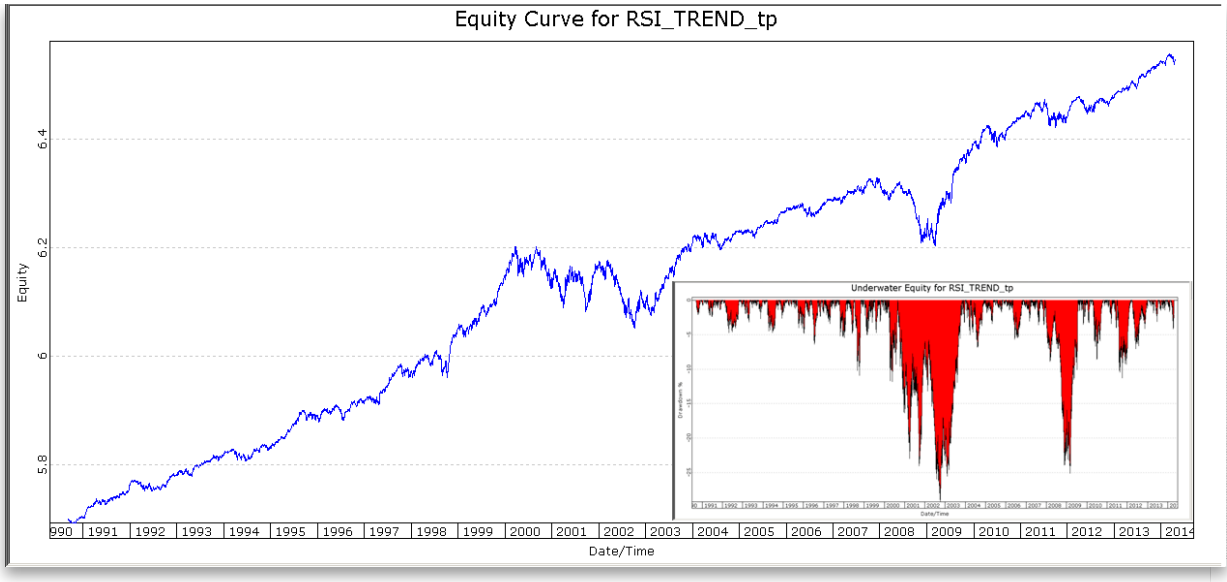


FIGURE 8: TRADERSSTUDIO. Here is the log equity curve and underwater equity curve for the RSITREND system run on the EQUALDOLLAR tradeplan using the NASDAQ 100 list of stocks for the years 1991–2014.

TradersStudio code:

```
'RSI & PRICE TRENDS

'Author: Kevin Luo, TASC June 2015

'Coded by: Richard Denning 4/22/2015

'www.TradersEdgeSystems.com


Sub RSITREND(rsiLen,buyLvl,LLlen1,LLlen2,minPctUp1,minPctUp2,exitLvl)

'rsiLen=14,buyLvl=50,LLlen1=250,LLlen2=20,minPctUp1=20,minPctUp2=5,exitLvl=70

Dim theRSI As BarArray

theRSI = rsi(C,rsiLen,0)

'Dim LL1 As BarArray

'Dim LL2 As BarArray

Dim LL1, LL2, barsSinceLL1, barsSinceLL2,  HHsinceLL1, HHsinceLL2, RoC1, RoC2

Dim trendUpLT As BarArray

Dim trendUpST As BarArray

LL1 = Lowest(L,LLlen1,0)

LL2 = Lowest(L,LLlen2,0)

barsSinceLL1 = MRO(L = LL1,LLlen1,1)
```

```
barsSinceLL2 = MRO(L = LL2,LLlen2,1)

HHsinceLL1 = Highest(H,barsSinceLL1,0)

HHsinceLL2 = Highest(H,barsSinceLL2,0)

RoC1 = (HHsinceLL1 / LL1 - 1) * 100

RoC2 = (HHsinceLL2 / LL2 - 1) * 100

trendUpLT = RoC1 >= minPctUp1

trendUpST = RoC2 >= minPctUp2

If trendUpLT And trendUpST And theRSI>buyLvl And theRSI[1]<= buyLvl Then

    Buy("LE",1,0,Market,Day)

End If

If RoC1 < minPctUp1 Then ExitLong("LX_trend","",1,0,Market,Day)

If theRSI<exitLvl And theRSI[1]>=exitLvl Then ExitLong("LX_rsi","",1,0,Market,Day)

End Sub
```

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

NEUROSHELL TRADER: JULY 2015

The SRSI, described by Vitali Apirine in his April 2015 article in S&C, "The Slow Relative Strength Index," can be easily implemented with a few of NeuroShell Trader's 800+ indicators. Simply select *new indicator* from the *insert* menu and use the indicator wizard to create the following indicators:

```
R1: ExpAvg(Close,6)
R4: ExpAvg(IfThenElse(A>B(Close,R1),Sub(Close,R1),0),14)
R5: ExpAvg(IfThenElse(A<B(Close,R1),Sub(R1,Close),0),14)
RR: IfThenElse(A=B(R5,0),100,Sub(100,Divide(100,Add2(1,Divide(R4,R5)))))
```

Users of NeuroShell Trader can go to the STOCKS & COMMODITIES section of the NeuroShell Trader free technical support website to download a copy of this or any previous Traders' Tips.

A sample chart is shown in Figure 9.

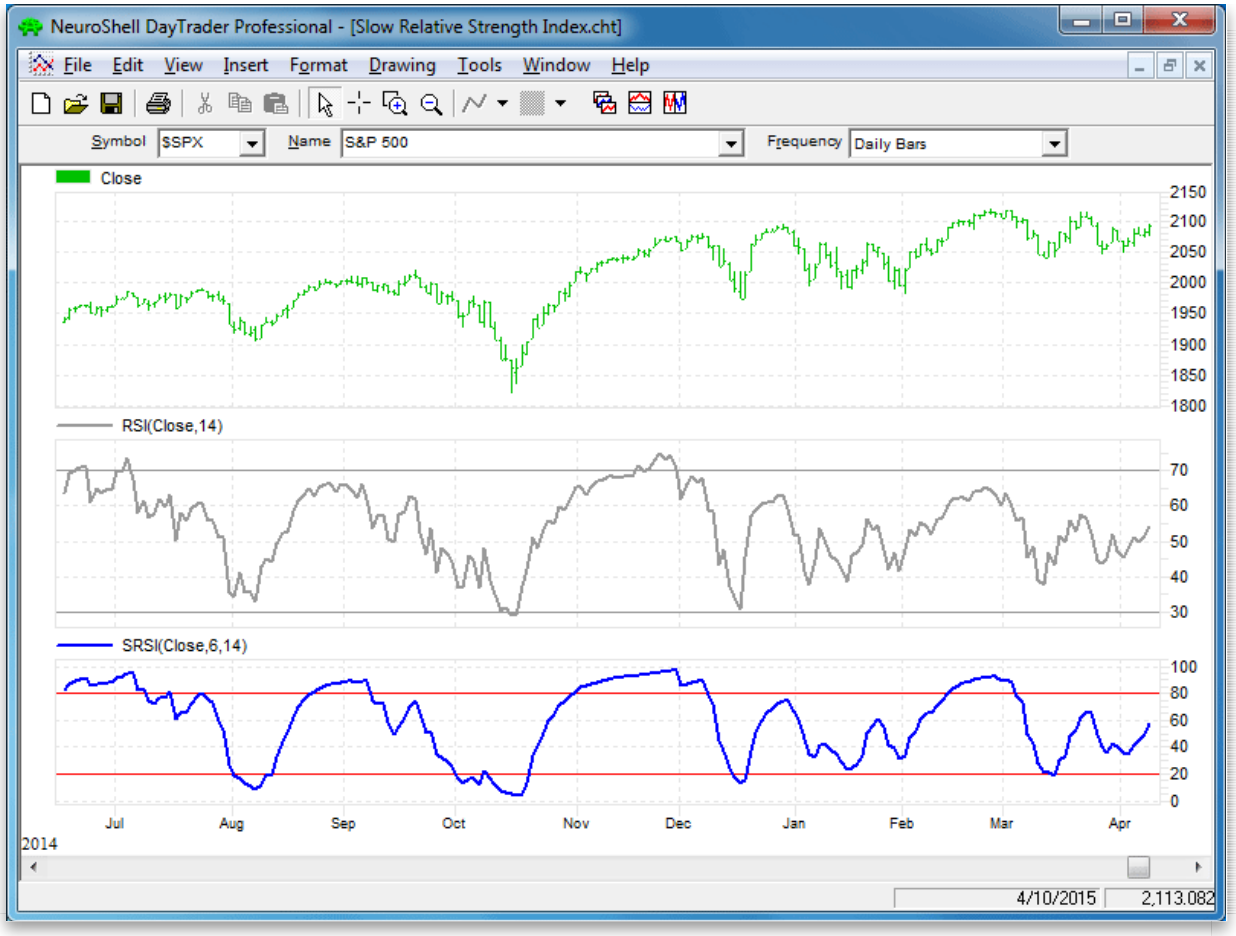


FIGURE 9: NEUROSHELL TRADER. This NeuroShell Trader chart displays the RSI and the SRSI on the S&P 500 index.

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

BACK TO LIST

NINJATRADER: JULY 2015

The SRSI or slow relative strength index, as discussed by Vitali Apirine in his April 2015 article in S&C, "The Slow Relative Strength Index," is available for download at www.ninjatrader.com/SC/July2015SC.zip.

Once it has been downloaded, from within the NinjaTrader Control Center window, select the menu File→Utilities→Import NinjaScript and select the downloaded file. This file is for NinjaTrader version 7 or greater.

You can review the indicator source code by selecting the menu Tools→Edit NinjaScript→Indicator from within the NinjaTrader Control Center window and selecting the "SRSI" file.

A sample chart implementing the indicator is shown in Figure 10.

FIGURE 10: NINJATRADER. This screenshot shows the SRSI applied to the S&P 500 index (SPX) in NinjaTrader from August 4, 2000 to October 2, 2000.

—Raymond Deux & Patrick Hodges
NinjaTrader, LLC
www.ninjatrader.com

BACK TO LIST

UPDATA: JULY 2015

This month's Traders' Tip is based on "The Slow Relative Strength Index" by Vitali Apirine, which appeared in the April 2015 issue of S&C.

The author delivers a variation on J. Welles Wilder's classic relative strength index (RSI) by substituting the close-to-close differences input, with the difference between the close to a moving average. This produces a smoother indicator (see Figure 11), and one the author suggests clarifies momentum changes and price–indicator divergences.

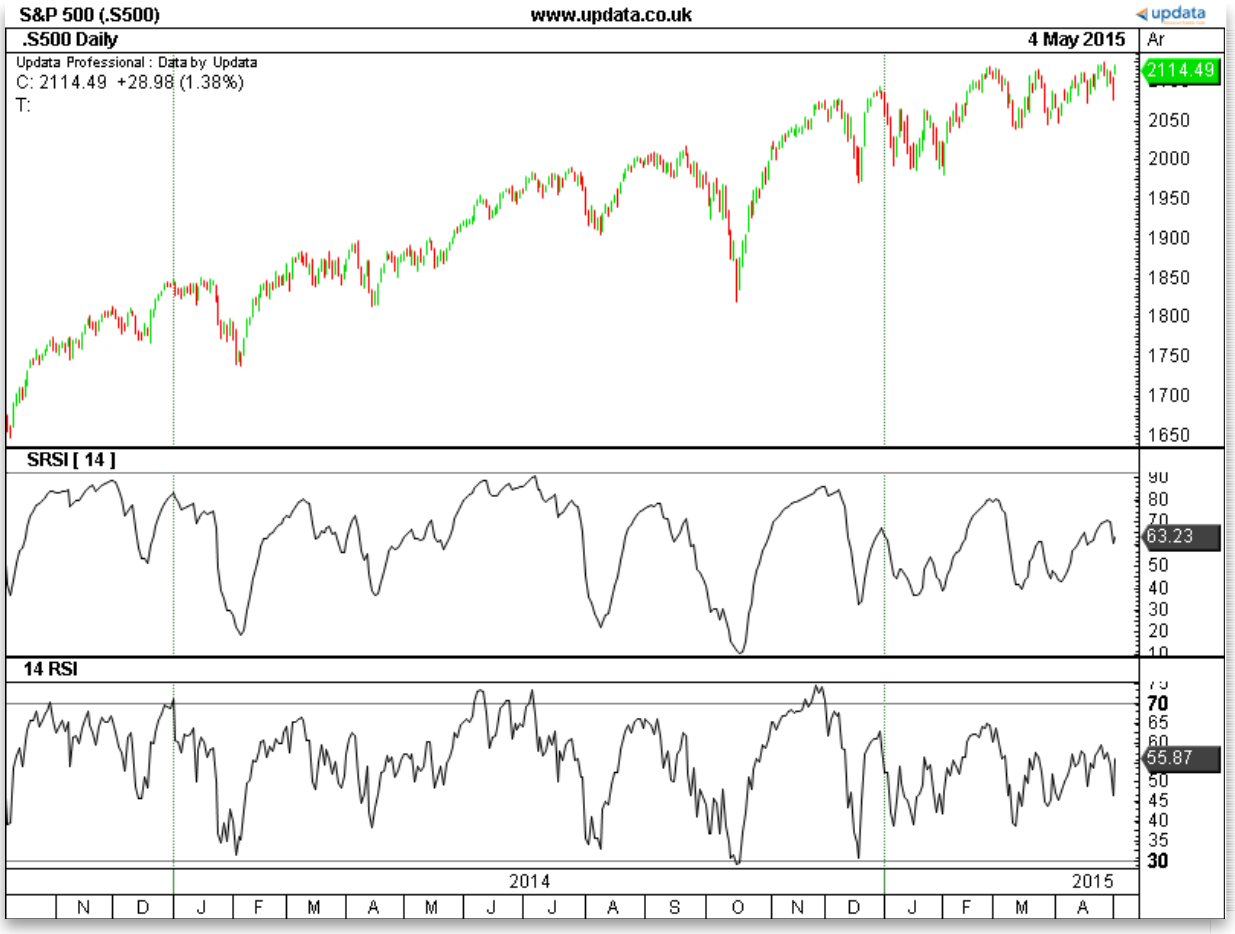


FIGURE 11: UPDATA. This chart shows the daily S&P 500 Index with the SRSI [14] shown with the standard RSI [14] for comparison.

The Updata code for this indicator is in the Updata library and may be dow nloaded by clicking the *custom* menu and then *indicator library*. The code can also be pasted into the Updata custom editor.

```
PARAMETER "Avg.Period" #PERIOD=6
PARAMETER "RSI Period" #RSIPeriod=14
NAME "SRSI [" #RSIPeriod "]" ""
@AVG=0
@POSITIVEDIFF=0
@NEGATIVEDIFF=0
@AVGPOSITIVEDIFF=0
@AVGNEGATIVEDIFF=0
@SRS=0
@SRSI=0
FOR #CURDATE=#PERIOD TO #LASTDATE
   @AVG=EAVE(#PERIOD)
   @POSITIVEDIFF=MAX(CLOSE-@AVG,0)
   @NEGATIVEDIFF=MAX(@AVG-CLOSE,0)
   IF #CURDATE=#PERIOD-1
      @AVGPOSITIVEDIFF=SGNL(@POSITIVEDIFF,#RSIPeriod,M)
      @AVGNEGATIVEDIFF=SGNL(@NEGATIVEDIFF,#RSIPeriod,M)
   ELSEIF #CURDATE>#PERIOD-1
      @AVGPOSITIVEDIFF=((#RSIPeriod-1)*@AVGPOSITIVEDIFF+@POSITIVEDIFF)/#RSIPeriod
      @AVGNEGATIVEDIFF=((#RSIPeriod-1)*@AVGNEGATIVEDIFF+@NEGATIVEDIFF)/#RSIPeriod
   ENDIF
```

```
    @SRS=@AVGPOSITIVEDIFF/@AVGNEGATIVEDIFF
    @SRSI=100-(100/(1+@SRS))
    @PLOT=@SRSI
  NEXT
```

—Updata support team
support@updata.co.uk, www.updata.co.uk

BACK TO LIST

MICROSOFT EXCEL: JULY 2015

"The Slow Relative Strength Index" by Vitali Apirine, which appeared in the April 2015 issue of S&C, discusses an indicator that is a variation of the relative strength index (RSI). The author uses the difference between the close and a moving average as the input. The slow relative strength index (SRSI) can be calculated using an Excel spreadsheet, as presented here.

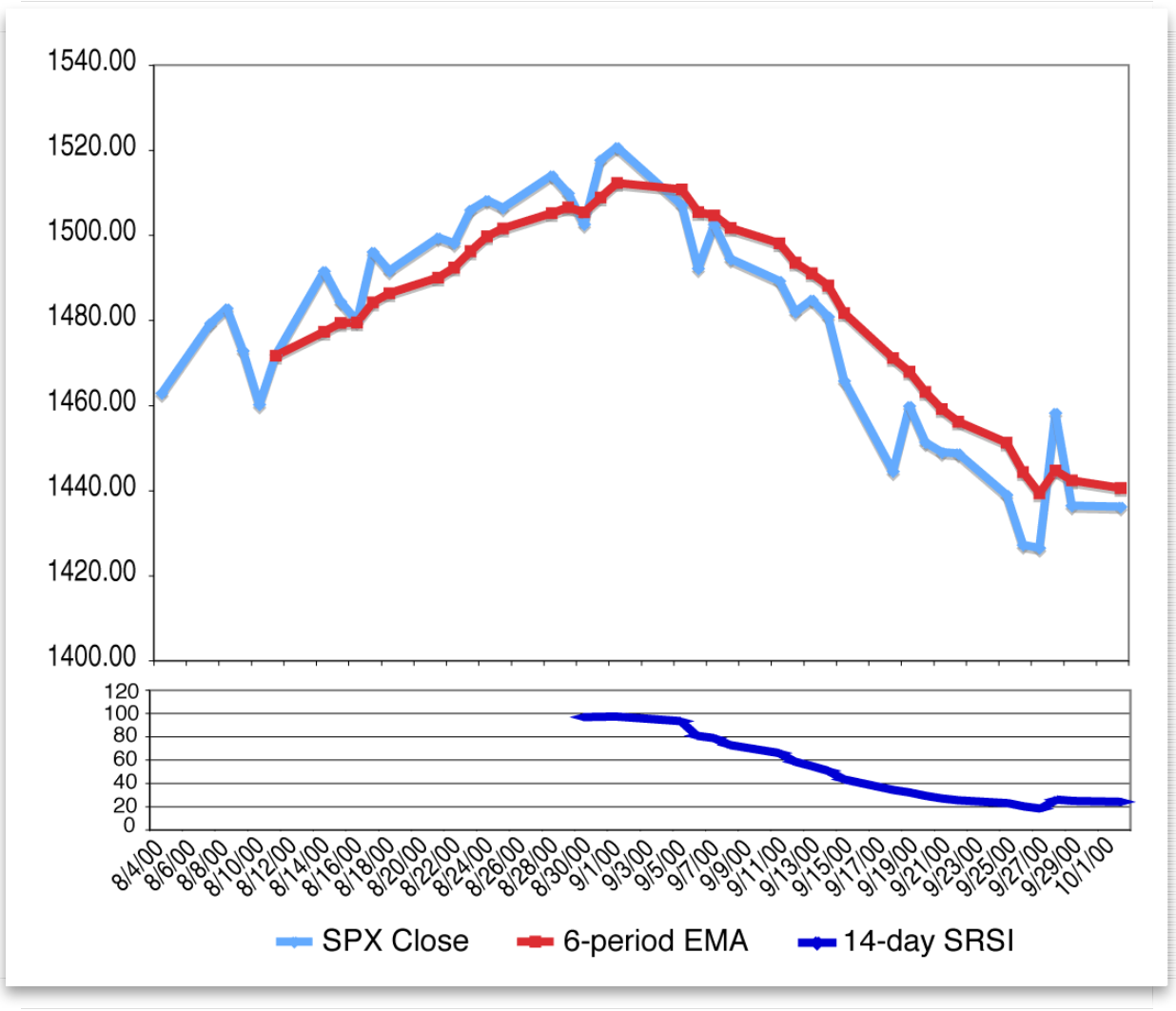


FIGURE 12: MICROSOFT Excel. This sample chart shows the SPX, a six-period EMA of the SPX, and the SRSI (6,14) in the subplot. The SRSI calculation begins after 14 days of data have been recorded.

The spreadsheet contains calculations for the SRSI (6,14), SRSI (8,14) and SRSI (8,17). A sample chart plotted from the calculations is shown in Figure 12. Following are the Excel spreadsheet formulas for the SRSI (6,14):

```
Column A: Date

Column B: SPX close (closing price of SPX).

Column C: 6-period exponential moving average (EMA) starting on row 7
=AVERAGE(B2:B7) and copy this down to all cells in column.

Column D: Positive difference starting on row 7
=IF(B7>C7,B7-C7,0) and copy this down to all cells in column.

Column E: Negative difference starting on row 7
=IF (B7<C7,C7-B7,0) and copy this down to all cells in column.

Column F: Average positive difference starting on row 20
=SUM(D7:D20)/14 and copy this down to all cells in column.

Column G: Average negative difference starting on row 20
=SUM(E7:E20)/14 and copy this down to all cells in column.

Column H: Slow relative strength (SRS) starting on row 20
=IF(G20=0,100,F20/G20) and copy this down to all cells in column.

Column I: 14-day slow relative strength index (SRSI) starting on row 20
=IF(H20=100,100,100-(100(1+H20)))
```

The calculations for SRSI (8,14) and SRSI (8,17) are similar except the eight-period EMA will be calculated after the eighth closing price and the SRSI calculations will start after 14 days of data have been recorded for SRSI (8,14) and after 17 days of data have been recorded for SRSI (8,17).

The spreadsheet file for this Traders' Tip can be downloaded here. To successfully download it, follow these steps:

- Right-click on the Excel file link, then
- Select "save as" (or "save target as") to place a copy of the spreadsheet file on your hard drive.

—Editor

BACK TO LIST

Originally published in the July 2015 issue of
Technical Analysis of STOCKS & COMMODITIES magazine.

August 2015

For this month's Traders' Tips, we focus on Vitali Apirine's June 2015 article, "The Slow Volume Strength Index," as a follow up to last month's Traders' Tips, which focused on Vitali Apirine's April 2015 article, "The Slow Relative Strength Index." Here, we present the August 2015 Traders' Tips code with possible implementations in various software.

The Traders' Tips section is provided to help the reader implement a selected technique from an article in this issue or another recent issue. The entries here are contributed by software developers or programmers for software that is capable of customization.

TRADESTATION: AUGUST 2015

In "The Slow Volume Strength Index," which appeared in the June 2015 issue of STOCKS & COMMODITIES, author Vitali Apirine describes a momentum volume oscillator called the *slow volume strength index* (SVSI) that he characterizes as measuring the buying and selling pressure relative to a price exponential moving average. He then uses a method of smoothing based on techniques originally presented by J. Welles Wilder. Apirine presents a number of ways to use the oscillator, including combining it with his *slow relative strength index* (SRSI) oscillator, which he presented in his article "The Slow Relative Strength Index," which appeared in the April 2015 issue of STOCKS & COMMODITIES.

For users' convenience, we are providing the code for a TradeStation SVSI indicator as well as simple strategy based on the author's work. A sample chart is shown in Figure 1.

```
{
TASC Jun 2015
The Slow Volume Strength Index
Indicator
Vitale Apirine
}
inputs:
        EMALength( 6 ),
        SmoothingLength( 14 ),
        OverBought( 80 ),
        OverSold( 20 ),
        MidLine( 50 ) ;
```

```
variables:
        EMAValue( 0 ),
        PosVolume( 0 ),
        NegVolume( 0 ),
        AvgPosVol( 0 ),
        AvgNegVol( 0 ),
        SVS( 0 ),
        SVSI( 0 ),
        MyVolume( 0 ) ;

if BarType >= 2 and BarType < 5 then
        MyVolume = Volume
else
        MyVolume = Ticks ;

EMAValue = XAverage( Close, EMALength ) ;

if Close > EMAValue then
        begin
        PosVolume = MyVolume ;
        NegVolume = 0 ;
        end
else if Close < EMAValue then
        begin
        PosVolume = 0 ;
        NegVolume = MyVolume ;
        end
else
        begin
        PosVolume = 0 ;
        NegVolume = 0 ;
        end ;

if CurrentBar = 1 then
        begin
        AvgPosVol = Average( PosVolume, SmoothingLength ) ;
        AvgNegVol = Average( NegVolume, SmoothingLength ) ;
        end
else if CurrentBar > 1 then
        begin
        AvgPosVol = ( ( AvgPosVol * 13 ) + PosVolume ) / 14 ;
        AvgNegVol = ( ( AvgNegVol * 13 ) + NegVolume ) / 14 ;
        end ;

SVS = iff( AvgNegVol <> 0, AvgPosVol / AvgNegVol, 100 ) ;
if SVS <> 0 then
        SVSI = 100 - ( 100 / ( 1 + SVS ) ) ;

Plot1( SVSI, "SVSI" ) ;
Plot2( OverBought, "OB" ) ;
Plot3( OverSold, "OS" ) ;
Plot4( MidLine, "MidLine" ) ;


Strategy SVSI

{
TASC Jun 2015
The Slow Volume Strength Index
```

```
Strategy
Vitale Apirine
}
inputs:
        EMALength( 6 ),
        SmoothingLength( 14 ),
        OverBought( 80 ),
        OverSold( 20 ),
        MidLine( 50 ) ;

variables:
        EMAValue( 0 ),
        PosVolume( 0 ),
        NegVolume( 0 ),
        AvgPosVol( 0 ),
        AvgNegVol( 0 ),
        SVS( 0 ),
        SVSI( 0 ),
        MyVolume( 0 ) ;

if BarType >= 2 and BarType < 5 then
        MyVolume = Volume
else
        MyVolume = Ticks ;

EMAValue = XAverage( Close, EMALength ) ;

if Close > EMAValue then
        begin
        PosVolume = MyVolume ;
        NegVolume = 0 ;
        end
else if Close < EMAValue then
        begin
        PosVolume = 0 ;
        NegVolume = MyVolume ;
        end
else
        begin
        PosVolume = 0 ;
        NegVolume = 0 ;
        end ;

if CurrentBar = 1 then
        begin
        AvgPosVol = Average( PosVolume, SmoothingLength ) ;
        AvgNegVol = Average( NegVolume, SmoothingLength ) ;
        end
else if CurrentBar > 1 then
        begin
        AvgPosVol = ( ( AvgPosVol * 13 ) + PosVolume ) / 14 ;
        AvgNegVol = ( ( AvgNegVol * 13 ) + NegVolume ) / 14 ;
        end ;

SVS = iff( AvgNegVol <> 0, AvgPosVol / AvgNegVol, 100 ) ;
if SVS <> 0 then
        SVSI = 100 - ( 100 / ( 1 + SVS ) ) ;

if SVSI crosses over MidLine then
        Buy next bar at Market
```

```
else if SVSI crosses under MidLine then
        SellShort next bar at Market ;
```

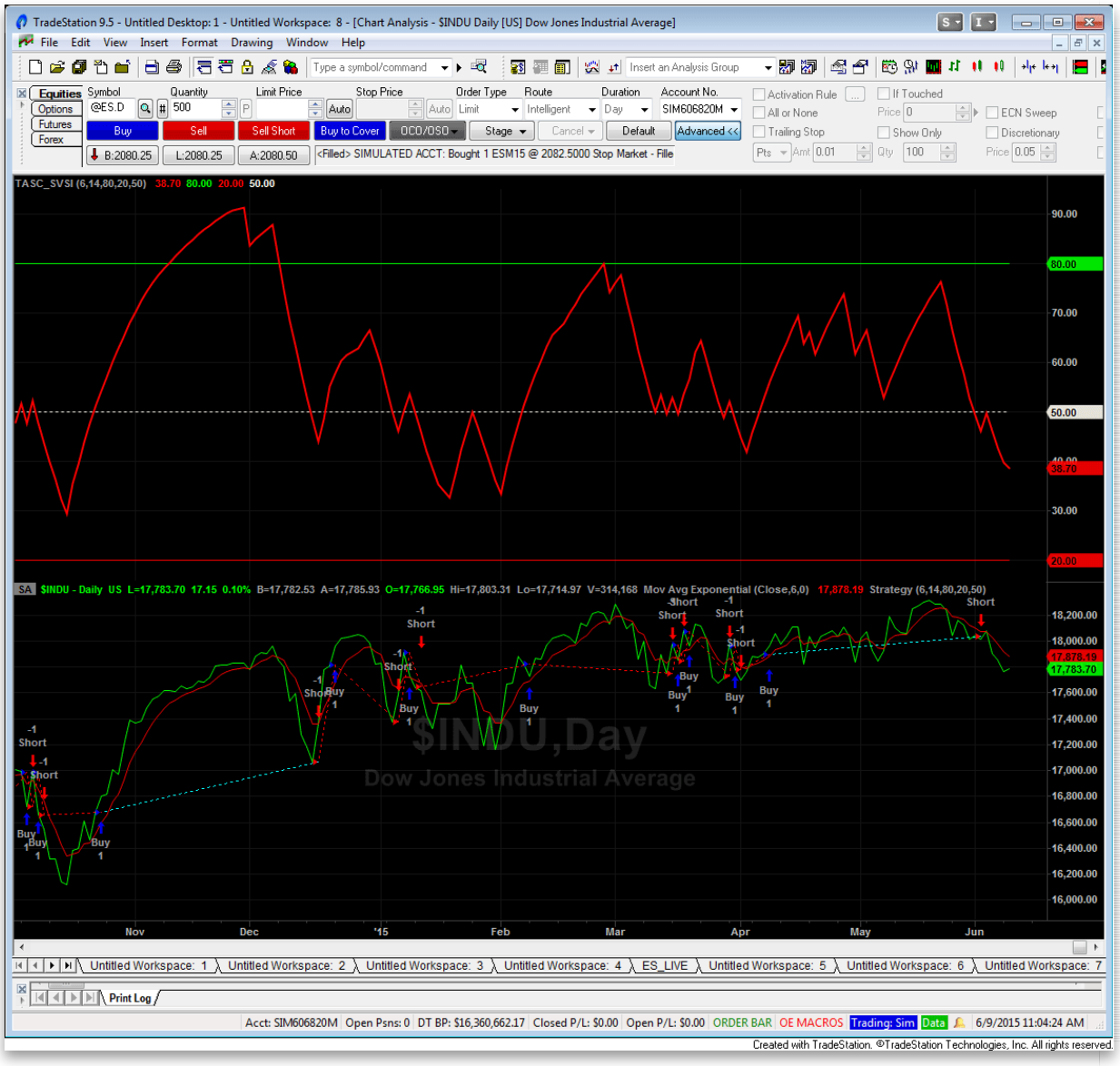


FIGURE 1: TRADESTATION. Here is an example of the SVSI indicator and strategy applied to a daily chart of the Dow Jones Industrial Average (DJIA).

To dow nload the EasyLanguage code, please visit our TradeStation and EasyLanguage support forum. The code for this article can be found here: http://w w w .tradestation.com/TASC-2015. The ELD filename is "TASC_AUG2015.ELD." For more information about EasyLanguage in general, please see http://w w w .tradestation.com/EL-FAQ.

This article is for informational purposes. No type of trading or investment recommendation, advice, or strategy is being made, given, or in any manner provided by TradeStation Securities or its affiliates.

—Doug McCrary
TradeStation Securities, Inc.
www.TradeStation.com

BACK TO LIST

eSIGNAL: AUGUST 2015

For this month's Traders' Tip, we've provided the study SVSI.efs based on the formula described in Vitali Apirine's article in the June 2015 issue of S&C, "The Slow Volume Strength Index." In it, Apirine presents a volume oscillator based on J. Welles Wilder's relative strength index (RSI). It also utilizes a price-based exponential moving average to determine overbought and oversold conditions, as well as divergences.

The study contains formula parameters that may be configured through the *edit chart* window (right-click on the chart and select "edit chart"). A sample chart plotting the SVSI is shown in Figure 2.

FIGURE 2: eSIGNAL. Here is an example of the SVSI study plotted on a daily chart of AAPL.

To discuss this study or download a complete copy of the formula code, please visit the EFS Library Discussion Board forum under the *forums* link from the support menu at www.esignal.com or visit our EFS KnowledgeBase at http://www.esignal.com/support/kb/efs/. The eSignal formula script (EFS) is also available here:

```
/*********************************
Provided By:
    Interactive Data Corporation (Copyright B© 2015)
    All rights reserved. This sample eSignal Formula Script (EFS)
    is for educational purposes only. Interactive Data Corporation
    reserves the right to modify and overwrite this EFS file with
    each new release.

Description:
    The Slow Volume Strength Index by Vitali Apirine

Formula Parameters:                     Default:
Length EMA                              6
Length Average Differences              14
Upper Bound                             80
Lower Bound                             20

Version:            1.00  06/05/2015

Notes:
    The related article is copyrighted material. If you are not a subscriber
    of Stocks & Commodities, please visit www.traders.com.

**********************************/

var fpArray = new Array();

function preMain(){

    setStudyTitle("SVSI");

    setCursorLabelName("Upper Bound", 0);
    setCursorLabelName("Lower Bound", 1);
    setCursorLabelName("Center Line", 2);
    setCursorLabelName("SVSI", 3);

    setDefaultBarFgColor(Color.grey, 0);
    setDefaultBarFgColor(Color.grey, 1);
    setDefaultBarFgColor(Color.grey, 2);

    setShowCursorLabel(false, 0);
    setShowCursorLabel(false, 1);
    setShowCursorLabel(false, 2);
    setShowCursorLabel(true, 3);

    setDefaultBarStyle(PS_SOLID, 0);
    setDefaultBarStyle(PS_SOLID, 1);
    setDefaultBarStyle(PS_DASHDOT, 2);
    setDefaultBarStyle(PS_SOLID, 3);

    var x = 0;

    fpArray[x] = new FunctionParameter("fpLengthEMA", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("Length EMA");
        setLowerLimit(1);
        setDefault(6);
    };

    fpArray[x] = new FunctionParameter("fpLengthAvgDiff", FunctionParameter.NUMBER);
```

```
    with(fpArray[x++]){
        setName("Length Average Differences");
        setLowerLimit(1);
        setDefault(14);
    };

    fpArray[x] = new FunctionParameter("fpSRSIHighBorder", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("Upper Bound");
        setLowerLimit(0);
        setUpperLimit(100);
        setDefault(80);
    };

    fpArray[x] = new FunctionParameter("fpSRSILowBorder", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("Lower Bound");
        setLowerLimit(0);
        setUpperLimit(100);
        setDefault(20);
    };
}

var bInit = false;
var bVersion = null;

var xDifferences = null;
var xPositiveDiff = null;
var xNegativeDiff = null;
var xPositiveDiffAvg = null;
var xNegativeDiffAvg = null;

function main(fpLengthEMA, fpLengthAvgDiff, fpSRSIHighBorder, fpSRSILowBorder){

    if (bVersion == null) bVersion = verify();
    if (bVersion == false) return;

    if (!bInit){

        xDifferences = efsInternal('Calc_Differences', fpLengthEMA);
        xPositiveDiff = getSeries(xDifferences, 0);
        xNegativeDiff = getSeries(xDifferences, 1);

        xPositiveDiffAvg = smma(fpLengthAvgDiff, xPositiveDiff);
        xNegativeDiffAvg = smma(fpLengthAvgDiff, xNegativeDiff);

        bInit = true;
    };

    var nPositiveDiffAvg = xPositiveDiffAvg.getValue(0);
    var nNegativeDiffAvg = xNegativeDiffAvg.getValue(0);

    if (nPositiveDiffAvg == null || nNegativeDiffAvg == null)
        return;

    var nSVSI = (nNegativeDiffAvg == 0) ? 100 : 100 - (100 / (1 + (nPositiveDiffAvg /
nNegativeDiffAvg)));

    return [fpSRSIHighBorder, fpSRSILowBorder, 50, nSVSI];
}
```

```
var xClose = null;
var xEMA = null;
var xVol = null;

function Calc_Differences(nLength){

    if (getBarState() == BARSTATE_ALLBARS){
        xClose = close();
        xEMA = ema(nLength);
        xVol = volume();
    }

    var nClose = xClose.getValue(0);
    var nEMA = xEMA.getValue(0);
    var nVol = xVol.getValue(0);

    if (nClose == null || nEMA == null || nVol == null)
        return;

    var nPositiveDiff = nClose > nEMA ? nVol : 0;
    var nNegativeDiff = nClose < nEMA ? nVol : 0;

    return [nPositiveDiff, nNegativeDiff];
}

function verify(){
    var b = false;
    if (getBuildNumber() < 3742){

        drawTextAbsolute(5, 35, "This study requires version 12.1 or later.",
            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,
            null, 13, "error");
        drawTextAbsolute(5, 20, "Click HERE to upgrade.@URL=http://www.esignal.com
/download/default.asp",
            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,
            null, 13, "upgrade");
        return b;
    }
    else
        b = true;

    return b;
}
```

—Eric Lippert
eSignal, an Interactive Data company
800 779-6555, www.eSignal.com

BACK TO LIST

In his June 2015 article "The Slow Volume Strength Index," author Vitali Apirine expands on the slow relative strength index (SRSI) concept presented in his April 2015 article, but this time, changes the focus to volume.

Keeping in the same theme as his earlier article that "slow and steady wins the race," he uses ideas from the SRSI concept but this time measures the relative strength of *volume*.

We have recreated his slow volume strength index (SVSI) as a thinkorswim study using our proprietary scripting language, *thinkscript*. We have made the loading process extremely easy—simply click on the link http://tos.mx/MEqL2n and choose *save script to thinkorswim*, then choose to rename your study to "Slow VSI." You can adjust the parameters of these within the *edit studies* window to fine-tune your variables.

FIGURE 3: THINKORSWIM. The SlowVSI is plotted on a chart of GE over the past year.

In the example shown in Figure 3, you see the Slow VSI plotted on a chart of GE over the past year. For a more detailed description of the indicator, please see Apirine's article in June 2015 issue.

—thinkorswim
A division of TD Ameritrade, Inc.
www.thinkorswim.com

BACK TO LIST

The slow relative volume index (SVSI) introduced by Vitali Apirine in his June 2015 STOCKS & COMMODITIES article, "The Slow Volume Strength Index," is a momentum volume oscillator similar to its sibling, the SRSI (slow relative strength index, described in Apirine's April 2015 STOCKS & COMMODITIES article, "The Slow Relative Strength Index") in application and interpretation. Oscillating between zero and 100, the SVSI becomes overbought after reaching 80 and oversold after dropping below 20. Signals can also be generated by looking for centerline crossovers and divergences.

As with the RSI (relative strength index) after which the SVSI is modeled, SVSI could be put to use to screen for dips in established trends. The following example trading system will focus on this ability. Here are its rules:

- ***Established long trend:*** Close price is above a medium-term SMA
- ***Dip in the uptrend:*** SVSI gets below 50
- **Enter long** at market next bar
- **Exit long** on a trailing stop when the low price falls below the lowest low of *N* days.

The trailing stop parameter (lowest low of *N* days) controls the trade duration: set it to a shorter period (five bars) to profit from small rallies, or to a longer period (for example, 20–50 bars) to catch larger trends (see Figure 4).

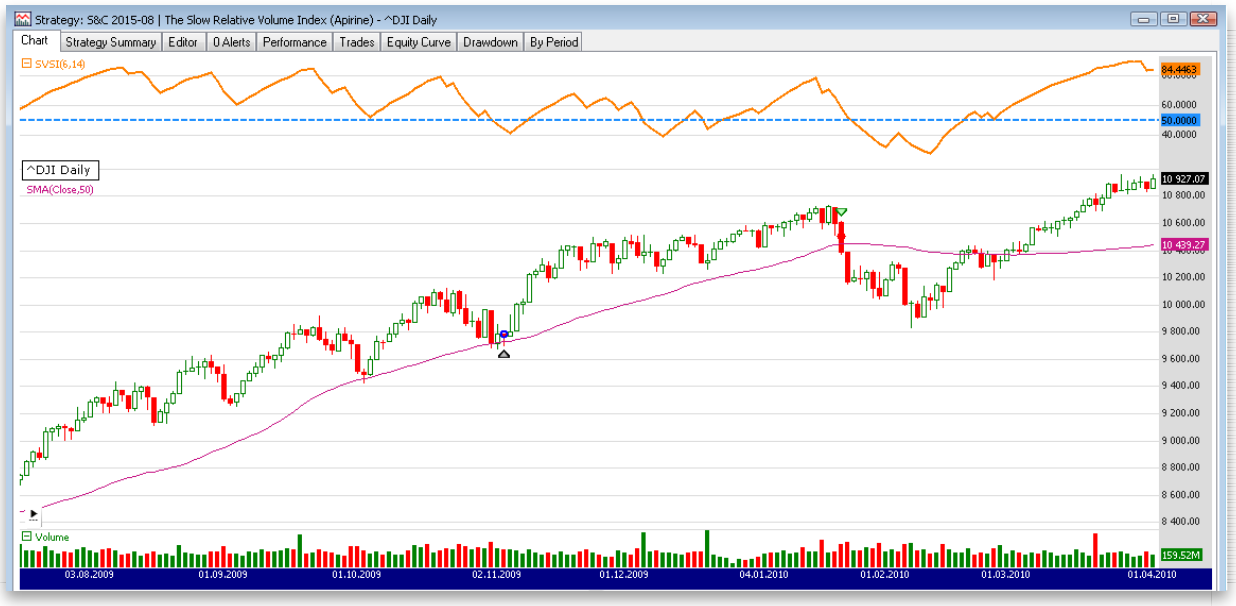


FIGURE 4: WEALTH-LAB, SVSI. Here is a characteristic trade on the ^DJIA when the SVSI dips below 50 in a strong uptrend.

You will find the SVSI indicator installed under the "TASC Magazine Indicators" group in Wealth-Lab after you update the TASCIndicators library to version 2015.07 or later, You can plot the indicator on a chart or use it as an entry or exit condition in a rule-based strategy without having to program any code yourself.

The code is shown below.

```
using System;
using System.Collections.Generic;
```

```
using System.Text;
using System.Drawing;
using WealthLab;
using WealthLab.Indicators;
using TASCIndicators;

namespace WealthLab.Strategies
{
        public class TASCTips201507 : WealthScript
        {
                private StrategyParameter paramSVSIPeriod;
                private StrategyParameter paramSVSIWMAPeriod;
                private StrategyParameter paramSVSIThreshold;
                private StrategyParameter paramSMAPeriod;
                private StrategyParameter paramExitChannel;

                public TASCTips201507()
                {
                        paramSVSIPeriod = CreateParameter("SVSI Period",6,6,30,2);
                        paramSVSIWMAPeriod = CreateParameter("SVSI WilderMA
Period",14,8,52,4);
                        paramSVSIThreshold = CreateParameter("SVSI
Threshold",50,40,90,5);
                        paramSMAPeriod = CreateParameter("SMA Period",40,10,100,10);
                        paramExitChannel = CreateParameter("Trailing Exit",20,5,50,5);
                }

                protected override void Execute()
                {
                        int svsiThreshold = paramSVSIThreshold.ValueInt;
                        int smaPeriod = paramSMAPeriod.ValueInt;
                        int trailingExit = paramExitChannel.ValueInt;
                        SVSI svsi = SVSI.Series(Bars,paramSVSIPeriod.ValueInt,
                                paramSVSIWMAPeriod.ValueInt);
                        SMA sma = SMA.Series(Close,smaPeriod);

                        ChartPane paneSVSI1 = CreatePane(40,true,true);

PlotSeries(PricePane,sma,Color.MediumVioletRed,LineStyle.Solid,1);

PlotSeries(paneSVSI1,svsi,Color.FromArgb(255,255,128,0),LineStyle.Solid,2);

DrawHorzLine(paneSVSI1,50,Color.DodgerBlue,LineStyle.Dashed,2);

                        for(int bar = GetTradingLoopStartBar(1); bar < Bars.Count;
bar++)
                        {
                                if (IsLastPositionActive)
                                {
                                        SellAtTrailingStop(bar+1,LastPosition,
                                                Lowest.Series(Low,trailingExit)[bar]);
                                }
                                else
                                {
                                        if( Close[bar] > sma[bar] &&
                                                svsi[bar] < svsiThreshold )
                                                        BuyAtMarket(bar+1);
                                }
                        }
                }
```

```
            }
  }
```

—Eugene, Wealth-Lab team
MS123, LLC
www.wealth-lab.com

BACK TO LIST

NEUROSHELL TRADER: AUGUST 2015

The slow volume strength index (SVSI) described by Vitali Apirine in his June 2015 article, “The Slow Volume Strength Index,” can be easily implemented with a few of NeuroShell Trader’s 800+ indicators. Simply select *new indicator* from the *insert* menu and use the indicator wizard to set up the following indicators:

```
C1: ExpAvg(Close,6)
V4: ExpAvg(IfThenElse(A>B(Close,C1),Volume,0),14)
V5: ExpAvg(IfThenElse(A<B(Close,C1),Volume,0),14)
RR: IfThenElse(A=B(V5,0),100,Sub(100,Divide(100,Add2(1,Divide(V4,V5)))))
```

Users of NeuroShell Trader can go to the STOCKS & COMMODITIES section of the NeuroShell Trader free technical support website to download a copy of this or any previous Traders’ Tips.

A sample chart is shown in Figure 5.

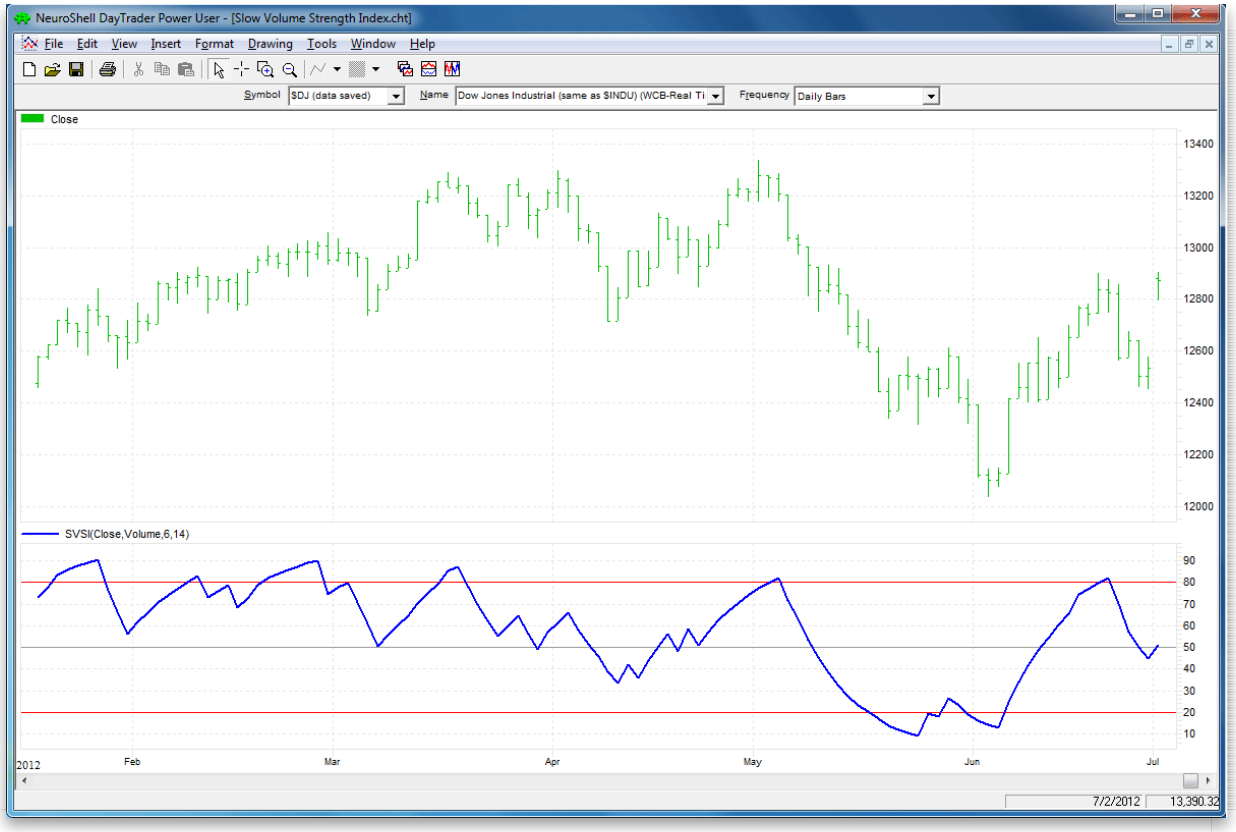


FIGURE 5: NEUROSHELL TRADER, SVSI. This sample NeuroShell Trader chart displays the SVSI on the DJIA.

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

BACK TO LIST

AIQ: AUGUST 2015

The AIQ code based on Vitali Apirine's June 2015 article in S&C, "The Slow Volume Strength Index," is provided for download from the following website:

- www.TradersEdgeSystems.com/traderstips.htm

```
!THE SLOW VOLUME STRENGTH INDEX
!Author: Vitali Aiprine, TASC April 2015
!Coded by: Richard Denning 6/10/2015
!www.TradersEdgeSystems.com

!INPUTS FOR INDICATOR:
```

```
emaLen is 6.
wilderLen is 14.

!INDICATOR FORMULAS:
ema is expavg([close],emaLen).
pDif is iff([close] - ema > 0,[volume],0).
nDif is iff([close] - ema < 0,[volume],0).

rsiLen  is 2 * wilderLen - 1.
AvgU    is expavg(pDif,rsiLen).
AvgD    is expavg(nDif,rsiLen).
svsi    is 100-(100/(1+(AvgU/AvgD))). !PLOT
```

The code provided for the slow volume strength index (SVSI) may be plotted as an indicator, as show n in Figure 6.

FIGURE 6: AIQ. Here is the SVSI (6,14) indicator compared to the classic RSI (14).

—Richard Denning
info@TradersEdgeSystems.com
for AIQ Systems

BACK TO LIST

TRADERSSTUDIO: AUGUST 2015

The TradersStudio code based on Vitali Apirine's June 2015 article in S&C, "The Slow Volume Strength

Index," is provided for download from the following website:

- www.TradersEdgeSystems.com/traderstips.htm

The following two code files are contained in the download:

- **Function SlowVRSI**—Computes the slow RSI values
- **Indicator plot slowVRSI_IND**—For plotting the slow RSI indicator on a chart.

FIGURE 7: TRADERSSTUDIO. Here is an example of the SVSI indicator on a chart of ESRX.

Figure 7 shows the indicator plotted on a chart of ESRX. The TradersStudio code is shown here:

```
'THE SLOW VOLUME STRENGTH INDEX
'Author: Vitali Aprine, TASC June 2015
'Coded by: Richard Denning 6/10/2015
'www.TradersEdgeSystems.com

'FUNCTION THAT RETURNS THE SVSI VALUES:
function SlowVRSI(emaLen, wilderLen)
'emaLen = 6, wilderLen = 14

'INDICATOR FORMULAS:
Dim ema,pDif,nDif,rsiLen,AvgU,AvgD
Dim sVSI As BarArray
ema = xaverage(C,emaLen)
pDif = IIF(C - ema > 0,V,0)
nDif = IIF(C - ema < 0,V,0)

rsiLen  = 2 * wilderLen - 1
AvgU    = xaverage(pDif,rsiLen)
AvgD    = xaverage(nDif,rsiLen)
if avgd <> 0 then sVSI = 100-(100/(1+(AvgU/AvgD)))
SlowVRSI = sVSI
End Function
'-------------------------------------------------------
```

```
'INDICATOR PLOT CODE:
sub SlowVRSI_IND(emaLen, rsiLen)
plot1(slowVRSI(emaLen, rsiLen))
plot2(70)
plot3(30)
End Sub
'-------------------------------------------------------
```

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

NINJATRADER: AUGUST 2015

We are providing a custom NinjaScript indicator based on Vitali Apirine's June 2015 article, "The Slow Volume Strength Index." This indicator, the *slow volume strength index* or SVSI, is available for dow nload at w w w .ninjatrader.com/SC/August2015SC.zip.

Once you have it dow nloaded, from w ithin the NinjaTrader Control Center w indow , select the menu File → Utilities → Import NinjaScript and select the dow nloaded file. This file is for NinjaTrader version 7 or greater.

You can review the indicator source code by selecting the menu Tools → Edit NinjaScript → Indicator from w ithin the NinjaTrader Control Center w indow and selecting the SVSI file.

NinjaScript uses compiled DLLs that run native, not interpreted, w hich provides you w ith the highest performance possible.

A sample chart implementing the indicator on a chart of the DJIA is show n in Figure 8.

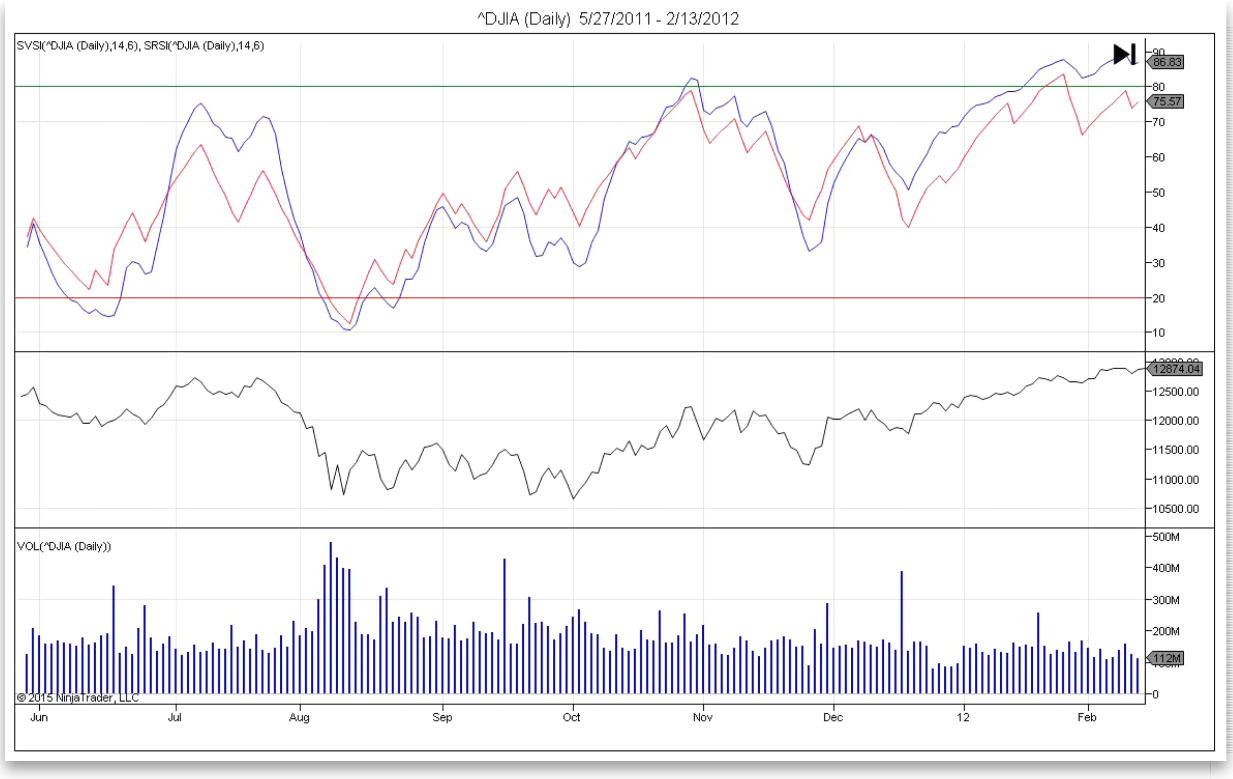


FIGURE 8: NINJATRADER. This chart shows the indicator applied to the daily DJIA index in NinjaTrader with the SVSI (slow volume strength index) and SRSI (slow RSI) plotted together above the DJIA. Volume is plotted below the DJIA.

—Raymond Deux & Patrick Hodges
NinjaTrader, LLC
www.ninjatrader.com

BACK TO LIST

UPDATA: AUGUST 2015

Our Traders' Tip for this month is based on an article by Vitali Apirine that appeared in the June 2015 issue of STOCKS & COMMODITIES titled "The Slow Volume Strength Index." In the article, Apirine delivers a variation of J. Welles Wilder's classic relative strength index (RSI). Apirine substitutes the close-to-close difference input in the RSI with the difference between the close and moving average as the input for the SVSI in order to assign positive and negative values to volume. See Figure 9 for an example implementation of the SVSI on a chart in Updata.

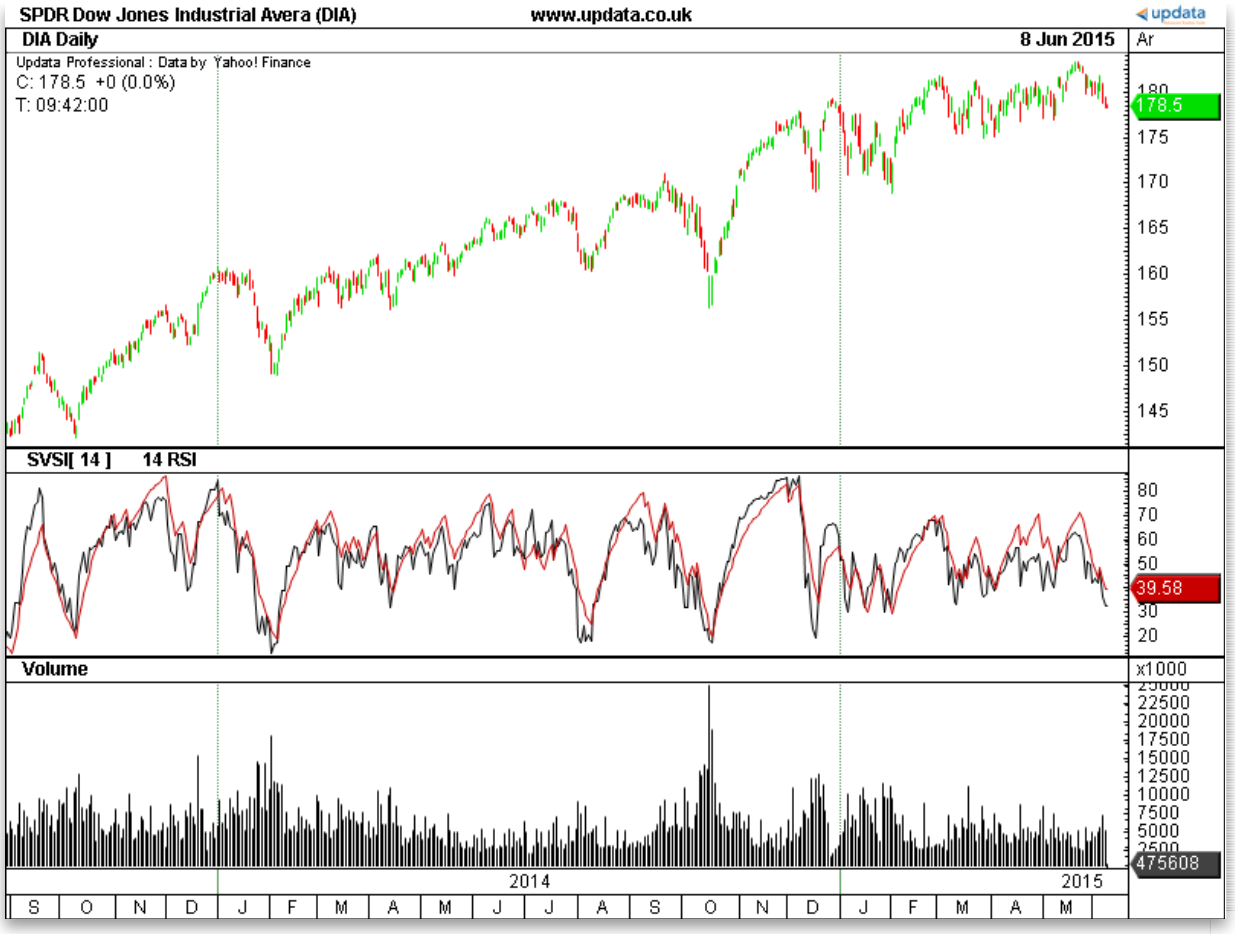


FIGURE 9: UPDATA, SVSI. This chart shows the Dow Jones Industrial Average ETF (DIA) with the SVSI [14] shown in red and the standard RSI [14] for comparison.

The Updata code for this indicator is in the Updata Library and it may be dow nloaded by clicking the *custom* menu and then *indicator library*. Those w ho cannot access the library due to a firew all may paste the code show n here into the Updata custom editor and save it.

```
PARAMETER "Smooth Period" #SMOOTH=6
PARAMETER "SVSI Period" #PERIOD=14
NAME "SVSI[" #PERIOD "]" ""
COLOUR RGB(200,0,0)
@AVG=0
@POSVOL=0
@NEGVOL=0
@SVSI=0
FOR #CURDATE=#PERIOD TO #LASTDATE
    @AVG=SGNL(CLOSE,#SMOOTH,E)
    IF #CURDATE<#PERIOD-1
       @POSVOL=SGNL(MAX(CLOSE>@AVG,0)*VOL,#PERIOD-1,M)
       @NEGVOL=SGNL(MAX(CLOSE<@AVG,0)*VOL,#PERIOD-1,M)
    ELSE
       @POSVOL=(MAX(CLOSE>@AVG,0)*VOL+(#PERIOD-1)*@POSVOL)/#PERIOD
       @NEGVOL=(MAX(CLOSE<@AVG,0)*VOL+(#PERIOD-1)*@NEGVOL)/#PERIOD
       @SVSI=100-(100/(1+(@POSVOL/@NEGVOL)))
    ENDIF
    @PLOT=@SVSI
```

NEXT

—Updata support team
support@updata.co.uk
www.updata.co.uk

BACK TO LIST

Originally published in the August 2015 issue of
Technical Analysis of STOCKS & COMMODITIES magazine.

October 2015

For this month's Traders' Tips, the focus is Vitali Apirine's article in this issue, "The Money Flow Oscillator." Here, we present the October 2015 Traders' Tips code with possible implementations in various software.

The Traders' Tips section is provided to help the reader implement a selected technique from an article in this issue or another recent issue. The entries here are contributed by software developers or programmers for software that is capable of customization.

TRADESTATION: OCTOBER 2015

In the article "The Money Flow Oscillator" in this issue, author Vitali Apirine presents a new indicator he developed as an alternative to existing methods for measuring buying and selling pressure. He describes this new money flow oscillator (MFO) as measuring the amount of money flow volume over a specific lookback period. A move into positive territory indicates buying pressure while a move into negative territory indicates selling pressure.

We are providing TradeStation EasyLanguage code for the MFO based on the article. In addition, we are providing the code for a TradeStation function to calculate the MFO and a strategy that demonstrates the function's usage.

```
Indicator: _MFO
inputs:
        Length( 20 ) ;

variables:
        Dvs( 0 ),
        MLTP( 0 ),
        MyVol( 0 ),
        Dvsv( 0 ),
        MFV( 0 ),
```

```
        MFO( 0 );

Dvs = ( High - Low[1] ) + ( High[1] - Low ) ;
if Dvs <> 0 then
        MLTP = Round(( ( High - Low[1] )
        - ( High[1] - Low ) ) / Dvs,2) ;

MyVol = iff( BarType >= 2 and
        BarType < 5, Volume, Ticks ) ;
Dvsv = iff ( MyVol <> 0, MyVol, 0 ) ;

MFV = ( MLTP * MyVol ) ;

MFO = Summation( MFV, Length ) /
        Summation( Dvsv, Length ) ;

Plot1( MFO, "MFO" ) ;
Plot2( 0, "ZL" ) ;
```

Function: _MFO

```
inputs:
        Length( NumericSimple ) ;

variables:
        Dvs( 0 ),
        MLTP( 0 ),
        MyVol( 0 ),
        Dvsv( 0 ),
        MFV( 0 ) ;

Dvs = ( High - Low[1] ) + ( High[1] - Low ) ;
if Dvs <> 0 then
        MLTP = Round(( ( High - Low[1] )
        - ( High[1] - Low ) ) / Dvs,2) ;

MyVol = iff( BarType >= 2 and
        BarType < 5, Volume, Ticks ) ;
Dvsv = iff ( MyVol <> 0, MyVol, 0 ) ;

MFV = ( MLTP * MyVol ) ;

_MFO = Summation( MFV, Length ) /
        Summation( Dvsv, Length ) ;
```

Strategy: _MFO

```
inputs:
        Length( 20 ),
        ConfirmBars( 2 ) ;

variables:
        UpCounter( 0 ),
        DnCounter( 0 ),
        MFOValue( 0 ) ;

MFOValue = _MFO( Length ) ;
```

```
if MFOValue > 0 then
        begin
        UpCounter += 1 ;
        DnCounter = 0 ;
        end
else
        begin
        UpCounter = 0 ;
        DnCounter += 1 ;
        end ;

if UpCounter >= ConfirmBars then
        Buy ( "MOF LE" ) next bar at Market
else if DnCounter >= ConfirmBars then
        SellShort ( "MOF SE" ) next bar at Market ;
```

To dow nload the EasyLanguage code, please visit our TradeStation and EasyLanguage support forum. The code for this article can be found at http://w w w .tradestation.com/TASC-2015. The ELD filename is "TASC_OCT2015.ELD."

For more information about EasyLanguage in general, please see http://w w w .tradestation.com/EL-FAQ.

A sample chart is show n in Figure 1.

FIGURE 1: TRADESTATION. In this example, the indicator and strategy based on the money flow oscillator are applied to a daily chart of Goldman Sachs (GS).

This article is for informational purposes. No type of trading or investment recommendation, advice, or strategy is being made, given, or in any manner provided by TradeStation Securities or its affiliates.

—Doug McCrary
TradeStation Securities, Inc.
www.TradeStation.com

BACK TO LIST

eSIGNAL: OCTOBER 2015

For this month's Traders' Tip, w e've provided the study MFO.efs based on the formula described in Vitali Apirine's article in this issue, "The Money Flow Oscillator." The author presents a study that measures buying and selling pressure.

The study contains formula parameters that may be configured through the edit chart w indow (right-click on the chart and select "edit chart"). A sample chart is show n in Figure 2.

To discuss this study or dow nload a complete copy of the formula code, please visit the EFS Library Discussion Board forum under the *forums* link from the support menu at w w w .esignal.com or visit our EFS Know ledgeBase at http://w w w .esignal.com/support/kb/efs/.

—Eric Lippert
eSignal, an Interactive Data company
800 779-6555, www.eSignal.com

BACK TO LIST

THINKORSWIM: OCTOBER 2015

In his article in this issue, “The Money Flow Oscillator,” Vitali Apirine discusses general indicators that measure buying and selling pressure and uses these same ideas to go in depth on his money flow oscillator (MFO). We have built a study for use in thinkorswim based on his ideas using our proprietary scripting language, thinkscript. We have made the loading process easy—simply click on the link http://tos.mx/e25ZJl and choose save script to thinkorswim, then choose to rename your study as “MoneyFlow Oscillator.” You can adjust the parameters within the edit studies window to fine-tune your variables.

FIGURE 3: THINKORSWIM. In this example, the MoneyFlowOscillator (MFO) is displayed below a chart of the Dow Jones Industrial Average from 2002.

In the example shown in Figure 3, you see the MoneyFlow Oscillator below a chart of the Dow Jones Industrial Average (DJIA) from 2002. You can see that as the MFO moves below the zero line, the DJIA price moves down drastically.

—thinkorswim
A division of TD Ameritrade, Inc.
www.thinkorswim.com

BACK TO LIST

WEALTH-LAB: OCTOBER 2015

The money flow oscillator (MFO) introduced by Vitali Apirine in his article in this issue, “The Money Flow

Oscillator," measures buying and selling pressure over a specific lookback. Bullish divergence indicates less selling pressure, and bearish divergence indicates less buying pressure. Signals can be generated by looking for divergences of MFO with price and centerline crossovers. For our example code, we will combine both ideas into a countertrend long-only trading system.

To identify divergences between price and oscillator, we'll be applying a straightforward approach. A divergence is detected when the SRSI indicator fails to confirm a price extreme, that is, the highest high of 20 days for bearish divergence or the 20-day lowest low for bullish divergence. This technique improves divergence detection time, practically reducing delay to a minimum compared to finding retracements from recent peaks or troughs.

System rules

1. Once a bullish divergence is detected, **enter long** next bar at open if MFO is below its centerline
2. **Exit long** next bar at open when MFO crosses above the centerline.

Trades from the short side are deliberately not taken, as their performance seems poor.

After updating the TASCIndicators library to v2015.09 or later, the MoneyFlow Oscillator indicator can be found under the TASC Magazine Indicators group. You can plot it on a chart or use it as an entry or exit condition in a rule-based strategy without having to program any code yourself.

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Drawing;
using WealthLab;
using WealthLab.Indicators;
using TASCIndicators;

namespace WealthLab.Strategies
{
        /*
                MFO divergence:
                Price sets a lowest low but the indicator fails to confirm the new low
        */

        public class MFO_Divergence : WealthScript
        {
                private StrategyParameter paramHighest;
                private StrategyParameter paramPeriod;

                public MFO_Divergence()
                {
                        paramPeriod = CreateParameter("MFO period", 20, 2, 100, 1);
                        paramHighest = CreateParameter("Highest high of", 20, 5, 50, 1)
                }

                protected override void Execute()
                {
                        bool peak = false; int peakBar = -1;
                        int high = paramHighest.ValueInt;
                        bool trough = false; int troughBar = -1;
                        int low = paramHighest.ValueInt;
                        int period = paramPeriod.ValueInt;

                        MoneyFlowOscillator mfo = MoneyFlowOscillator.Series( Bars, per
```

```
                        Lowest indicatorLowest = Lowest.Series( mfo, low );
                        Lowest hLow = Lowest.Series( Low, low );

                        HideVolume(); LineStyle solid = LineStyle.Solid;
                        ChartPane mfoPane = CreatePane( 50, false, true );
                        PlotSeries( mfoPane, mfo, Color.Green, solid, 2 );
                        DrawHorzLine( mfoPane,0,Color.Blue,LineStyle.Dashed,1);

                        for(int bar = GetTradingLoopStartBar(period); bar < Bars.Count,
                        {
                                if (!IsLastPositionActive)
                                {
                                        /* 1st peak: both price and indicator */

                                        if( peak == false )
                                        {
                                                if( ( High[bar-1] == Highest.Series( H:
                                                        & ( mfo[bar-1] == Highest.Serie
                                                        & TurnDown( bar, High ) & TurnI
                                                {
                                                        peak = true; peakBar = bar-1;
                                                }
                                        }

                                        if( peak == true )
                                        {
                                                if( ( High[bar] != Highest.Series( Hig
                                                        & ( mfo[bar] == Highest.Series
                                                        peak = false;
                                        }

                                        /* 2nd peak: price high not confirmed by the i

                                        if( peak == true )
                                        {
                                                if( ( High[bar-1] == Highest.Series( H:
                                                        & ( High[bar-1] >= High[peakBa
                                                        & ( mfo[bar-1] != Highest.Serie
                                                        & ( mfo[bar-1] < mfo[peakBar] )
                                                        TurnDown( bar, High ) & TurnDo
                                                {
                                                        peak = false;

                                                        /* Shorting doesn't work well

                                                        // Fade the trend

//                                                      if( mfo[bar] > 0 )
//                                                              if( ShortAtMarket( bar-
//                                                                      LastPosition.P

                                                        /* Highlight divergence */

                                                        for (int b = peakBar; b <= bar,
                                                                SetPaneBackgroundColor

                                                        DrawLine( PricePane, peakBar, I
                                                        DrawLine( mfoPane, peakBar, mf
                                                }
                                        }
```

```
                /* 1st trough: both price and indicator */

                if( trough == false )
                {
                        if( ( Low[bar-1] == Lowest.Series( Low,
                                & ( mfo[bar-1] == Lowest.Series
                                & TurnUp( bar, Low ) & TurnUp(
                        {
                                trough = true; troughBar = bar-
                        }
                }

                if( trough == true )
                {
                        if( ( Low[bar] != Lowest.Series( Low, 1
                                & ( mfo[bar] == Lowest.Series(
                                trough = false;
                }

                /* 2nd trough: price low not confirmed by the 1

                if( trough == true )
                {
                        if( ( Low[bar-1] == Lowest.Series( Low,
                                & ( Low[bar-1] <= Low[troughBar
                                & ( mfo[bar-1] != Lowest.Series
                                & ( mfo[bar-1] > mfo[troughBar]
                                TurnUp( bar, Low ) & TurnUp( ba
                        {
                                trough = false;

                                /* Fade the trend */

                                if( mfo[bar] < 0 )
                                        if( BuyAtMarket( bar+1
                                                LastPosition.Pr

                                /* Highlight divergence */

                                for (int b = troughBar; b <= ba
                                        SetPaneBackgroundColor
                                                Color.FromArgb

                                DrawLine( PricePane, troughBar,
                                        bar-1, Low[bar-1], Colo
                                DrawLine( mfoPane, troughBar, r
                                        bar-1, mfo[bar-1], Colo
                        }
                }
        } else
        {
                Position p = LastPosition;

                if( p.PositionType == PositionType.Long )
                {
                        if( CrossOver(bar, mfo, 0) )
                                ExitAtMarket( bar+1, p, "MFO Cr
                }
                else
```

```
                                                    if( CrossUnder(bar, mfo, 0) )
                                                            ExitAtMarket( bar+1, p, "MFO C
                                        }
                                }
                        }
                }
        }
```

A sample chart is show n in Figure 4.

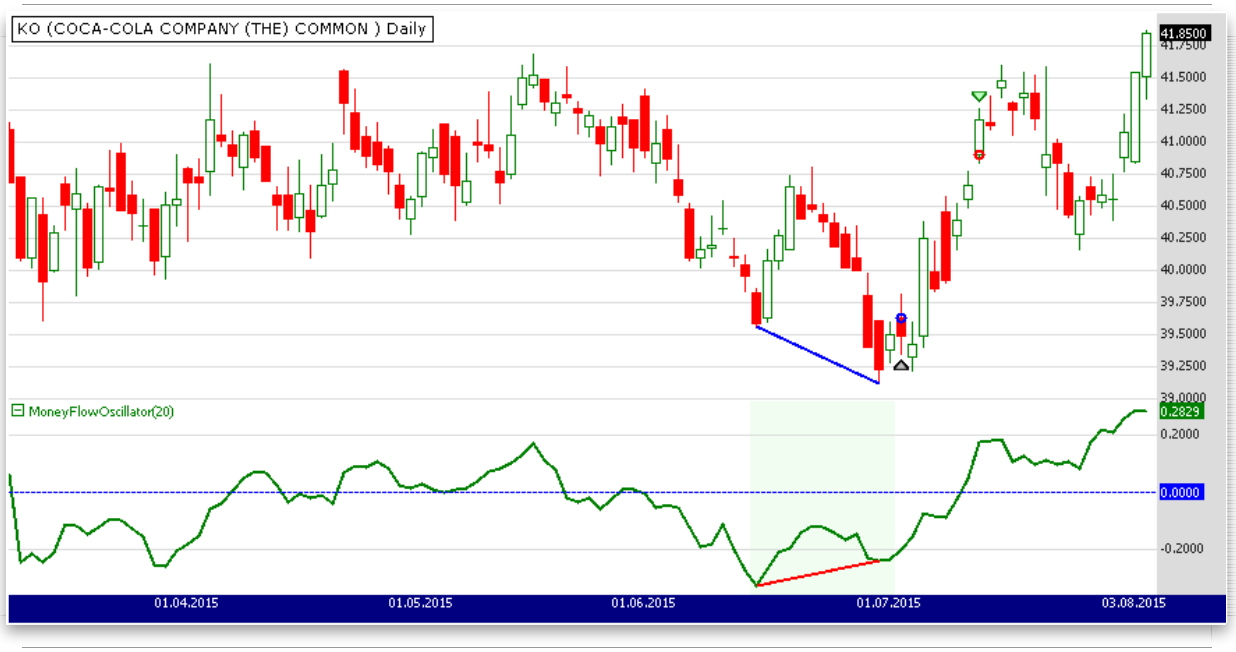


FIGURE 4: WEALTH-LAB. The bullish divergence between the MFO and price that formed in June 2015 triggered a long trade in KO (Coca Cola).

—Eugene, Wealth-Lab team
MS123, LLC
www.wealth-lab.com

BACK TO LIST

AMIBROKER: OCTOBER 2015

In “The Money Flow Oscillator” in this issue, author Vitali Apirine presents his money flow oscillator. A ready-to-use formula for AmiBroker based on the article is show n here. To use the oscillator, enter the code into the formula editor and press *apply indicator*. You can adjust the smoothing period using the parameters w indow .

```
// Money Flow Oscillator
LL = Low;
HH = High;

PL = Ref( LL, -1 );
PH = Ref( HH, -1 );
```

```
Dvs = ( HH - PL ) + ( PH - LL ) + 1e-10;

MLTP = IIf( HH < PL, -1,
       IIf( LL > PH,  1,
          ( ( HH - PL ) - ( PH - LL ) ) / dvs ) );

Dvsv = IIf( V == 0, 1e-10, V );

Periods = Param("Periods", 20, 1, 100 );

MFO = Sum( MLTP * V, Periods ) / Sum( Dvsv, Periods );

Plot( MFO, "MFO" + _PARAM_VALUES(), colorRed, styleThick );
Plot( 0, "", colorBlue );
```

A sample chart is show n in Figure 5.

FIGURE 5: AMIBROKER. Here is a monthly price chart of the Dow Jones Industrial Average (DJIA) with a 10-bar exponential moving average (middle pane) and a 20-bar money flow oscillator (upper pane), replicating the chart from Apirine's article in this issue.

—*Tomasz Janeczko, AmiBroker.com*
www.amibroker.com

BACK TO LIST

NEUROSHELL TRADER: OCTOBER 2015

The money flow oscillator described by Vitali Apirine in his article in this issue can be easily implemented in NeuroShell Trader with a few of NeuroShell Trader's 800+ built-in indicators. Simply select *new indicator* from the *insert* menu and use the indicator wizard to set up the following indicators:

```
HL1:    Subtract( High, Lag(Low,1) )
HL2:    Subtract( Lag(High,1), Low )
OSC:    Divide( Subtract(HL1,HL2), Add2(HL1,HL2) )
MLTP:   IfThenElse( A<B(High,Lag(Low,1) ), -1, A<B(Lag(High,1),Low), 1, OSC )
MFO:    Divide( Sum( Multiply2( MLTP, Volume ), 20), Sum( Volume, 20 ) )
```

Users of NeuroShell Trader can go to the STOCKS & COMMODITIES section of the NeuroShell Trader free technical support website to download a copy of this or any previous Traders' Tips.

A sample chart is shown in Figure 6.

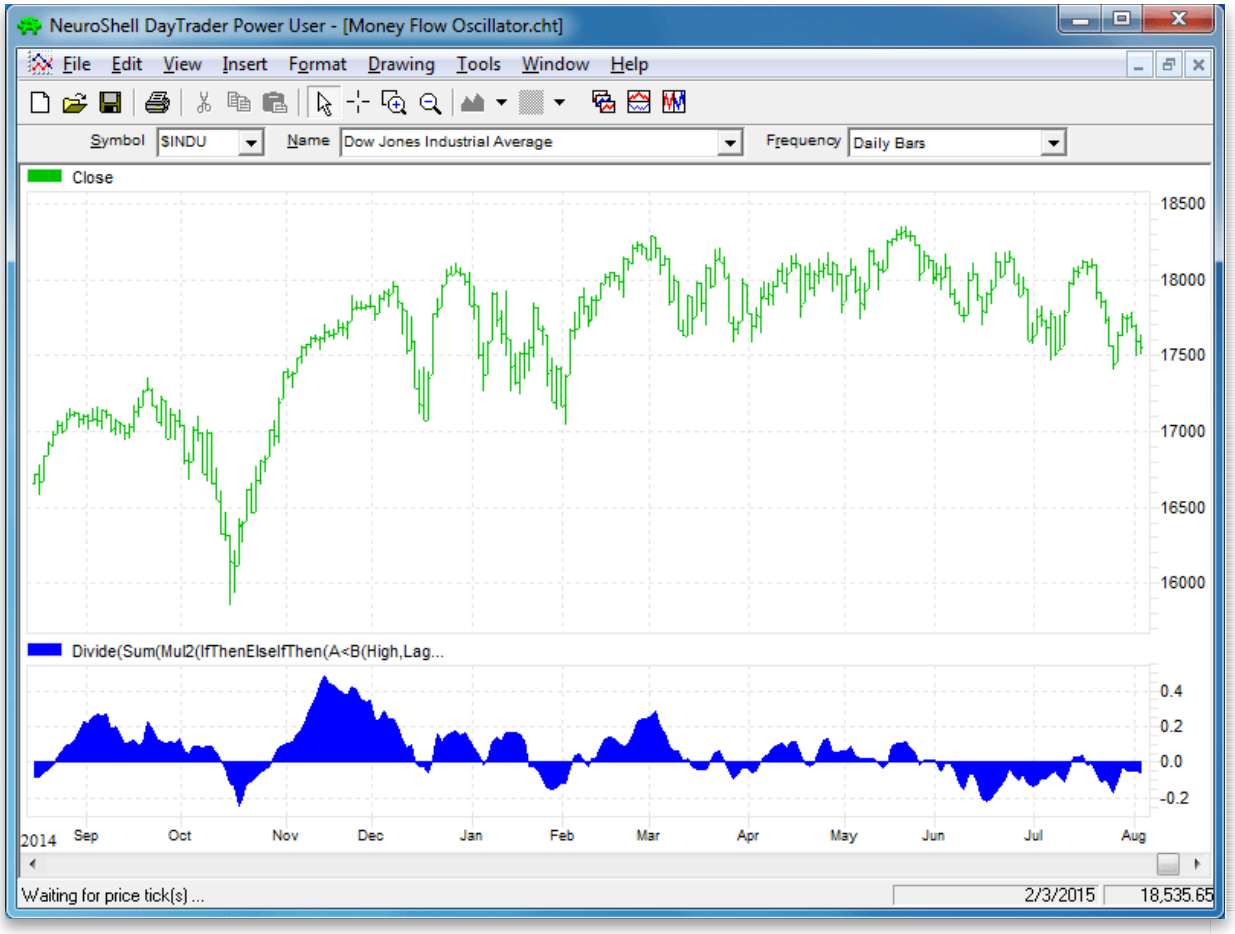


FIGURE 6: NEUROSHELL TRADER. This NeuroShell Trader chart displays the money flow oscillator on the DJIA.

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

BACK TO LIST

AIQ: OCTOBER 2015

The AIQ code I am providing here is based on Vitali Apirine's article in this issue, "The Money Flow Oscillator."

This code displays an indicator based on Apirine's money flow oscillator (MFO), an example of which is shown in Figure 7.

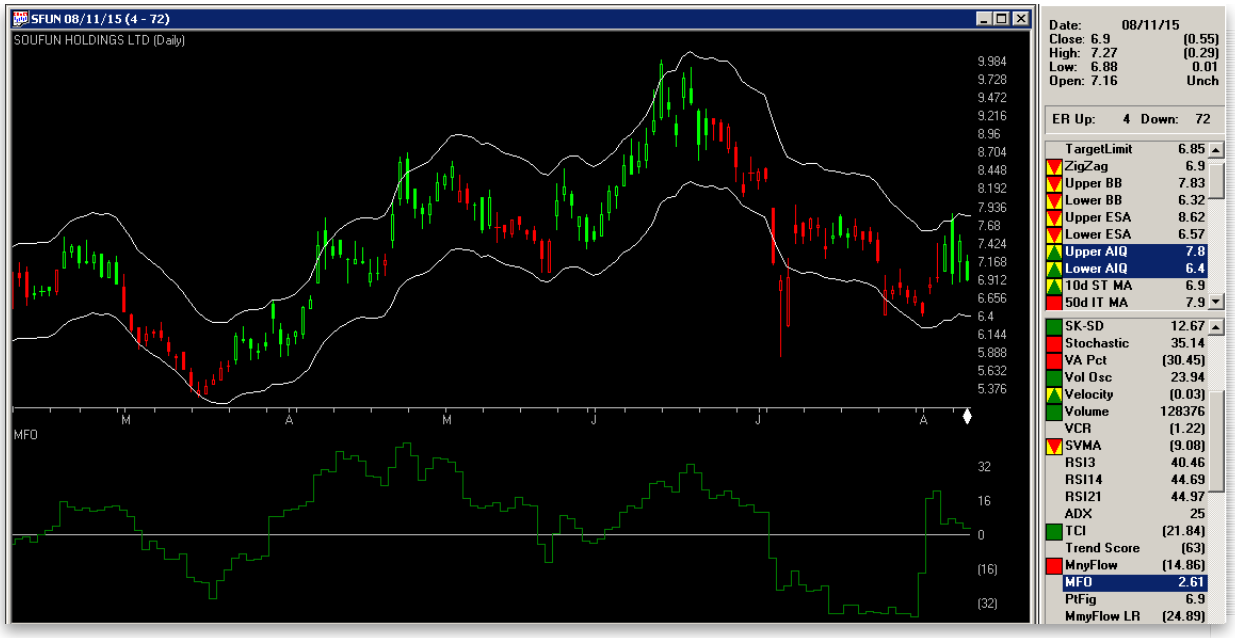


FIGURE 7: AIQ. Here is an example of the MFO(20) indicator on a chart of SFUN.

The code and EDS file can be dow nloaded from w w w .TradersEdgeSystems.com/traderstips.htm and is also show n here:

```
!THE MONEY FLOW OSCILLATOR
!Author: Vitali Aprine, TASC October 2015
!Coded by: Richard Denning, 8/11/15
!www.TradersEdgeSystems.com

!INPUTS:
MFOlen is 20.
H is [high].
H1 is valresult(H,1).
L is [low].
L1 is valresult([low],1).
V is [volume].

!INDICATOR CODE:
MFmult is ((H-L1)-(H1-L))/((H-L1)+(H1-L)).
MFvol is MFmult*V.
MFO is sum(MFvol,MFOlen)/sum(V,MFOlen)*100. !PLOT
```

—Richard Denning
info@TradersEdgeSystems.com
for AIQ Systems

BACK TO LIST

TRADERSSTUDIO: OCTOBER 2015

The TradersStudio code based on Vitali Apirine's article in this issue, "The Money Flow Oscillator," can be found at www.TradersEdgeSystems.com/traderstips.htm.

The following code files are provided in the download:

- Function MFO: Computes the MFO values
- Indicator plot MFO_IND: For plotting the MFO indicator on a chart.

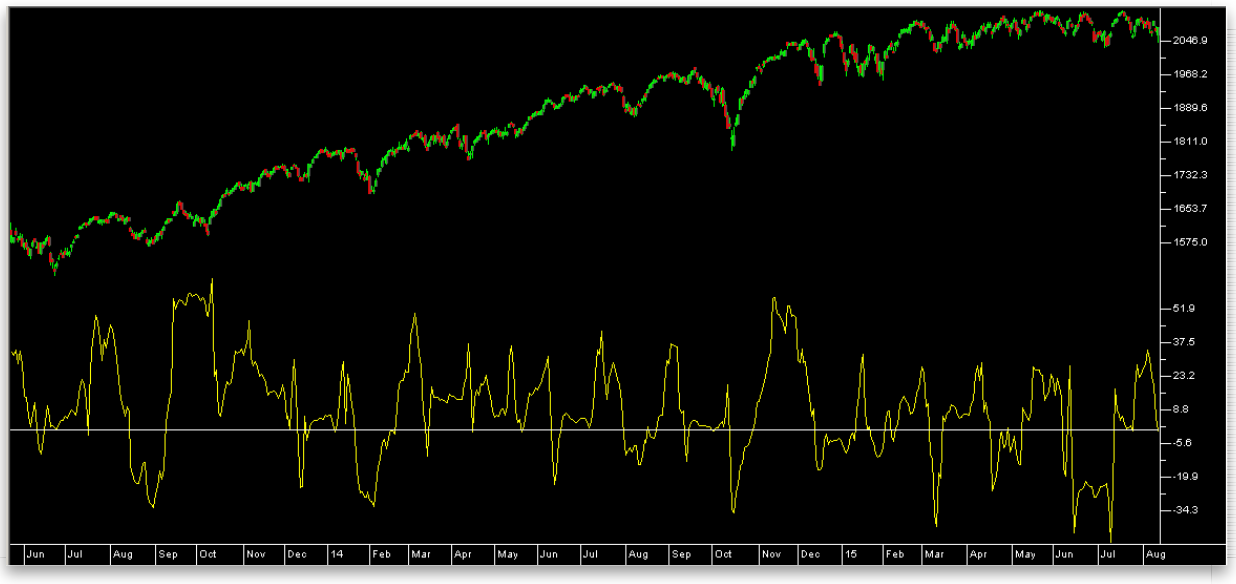


FIGURE 8: TRADERSSTUDIO. On this sample chart, the MFO indicator is plotted on a chart of S&P 500 futures contract (SP).

Figure 8 shows the MFO indicator on a chart of the S&P 500 futures contract (SP) using data from Pinnacle Data Corp.

The TradersStudio code is also shown here:

```
'THE MONEY FLOW OSCILLATOR
'Author: Vitali Aprine, TASC October 2015
'Coded by: Richard Denning, 8/11/15
'wwwTradersEdgeSystemscom

'FUNCTION TO COMPUTE MFO VALUES:
Function MFO(MFOlen)
'default: MFOlen = 20

Dim MFmult, MFOdenom
Dim MFsum As BarArray
Dim MFvol As BarArray

'INDICATOR CODE:
MFOdenom = ((H-L[1])+(H[1]-L))
If MFOdenom = 0 Then
    MFOdenom = 0.0000001
End If
```

```
MFmult = ((H-L[1])-(H[1]-L))/MFOdenom
MFvol = MFmult*V
MFsum = summation(V,MFOlen)
If MFsum = 0 Then
    MFsum = 0.000001
End If
MFO = Summation(MFvol,MFOlen)/MFsum*100
End Function
'-----------------------------------------
'INDICATOR CODE TO PLOT MFO INDICATOR:
sub MFO_IND(MFOlen)
plot1(MFO(MFOlen))
plot2(0)
end sub
'-----------------------------------------
```

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

NINJATRADER: OCTOBER 2015

The money flow oscillator presented by Vitali Apirine in his article in this issue by the same name has been made available for dow nload at w w w .ninjatrader.com/SC/October2015SC.zip.

Once you have it dow nloaded, from w ithin the NinjaTrader Control Center w indow , select the menu File → Utilities → Import NinjaScript and select the dow nloaded file. This file is for NinjaTrader version 7.

You can review the indicator's source code by selecting the menu Tools → Edit NinjaScript → Indicator from w ithin the NinjaTrader Control Center w indow and selecting the MFO file.

A sample chart displaying the indicator is show n in Figure 9.

FIGURE 9: NINJATRADER. The MFO is displayed on a chart of the DJIA along with volume and a 10-period EMA.

—Raymond Deux & Michael Maloney
NinjaTrader, LLC
www.ninjatrader.com

BACK TO LIST

UPDATA: OCTOBER 2015

Our Traders' Tip for this month is based on the article by Vitali Apirine in this issue, "The Money Flow Oscillator."

In his article, the author develops an oscillator that signals changes in buying or selling pressure for an instrument. He develops ratios betw een current and previous, high, and low prices, and then sums the volume over a predefined lookback period to produce an oscillator that is positive w hen buying pressure is detected and negative w hen selling pressure is detected.

FIGURE 10: UPDATA. Here, the money flow oscillator [20] is applied to the Dow Jones Industrial Average of daily resolution.

The Updata code for this article has been added to the Updata library and may be dow nloaded by clicking the *custom* menu and then *indicator library*. Those w ho cannot access the library due to a firew all may paste the code show n here into the Updata custom editor and save it.

```
'MFO
PARAMETER "Period" #PERIOD=20
NAME "MFO [" #PERIOD "]" ""
@MULT=0
@SUMVOL=0
@MFV=0
@MFO=0
FOR #CURDATE=#PERIOD TO #LASTDATE
   @MULT=((HIGH-LOW(1))-(HIGH(1)-LOW)) /((HIGH-LOW(1))+(HIGH(1)-LOW))
   @SUMVOL=SGNL(VOL,#PERIOD,M)*#PERIOD
   @MFV=@MULT*@SUMVOL
   @MFO=SGNL(@MFV,#PERIOD,M)*#PERIOD/@SUMVOL
   @PLOT=@MFO
NEXT
```

—Updata support team
support@updata.co.uk
www.updata.co.uk

BACK TO LIST

TRADE NAVIGATOR: OCTOBER 2015

We have prepared a custom file to make it easy to dow nload into Trade Navigator the indicator library based on “The Money Flow Oscillator” by Vitali Apirine in this issue. The file name is “SC201510.”

Please click on Trade Navigator’s blue telephone button, select *download special file*, then erase the w ord “upgrade” and type in “SC201508” (w ithout the quotes), and click the start button. When prompted to upgrade, click the yes button. If prompted to close all softw are, click on the *continue* button. Your library w ill now dow nload. This library contains an indicator named “MFO.” This indicator can be inserted into your chart by opening the *charting* dropdow n menu, then selecting the *add to chart* command, then selecting the *indicators* tab. This library also contains a template called “S&C Money Flow Oscillator.” This prebuilt template can be overlaid onto your chart by opening the *charting* dropdow n menu, then selecting the *templates* command, then selecting the template.

The TradeSense language for the indicator is as follow s:

```
&dvs := IFF ((High - Low.1) + (High.1 - Low) = 0 , 0.00001 , (High - Low.1) + (High.1
- Low))
&mltp := IFF (High < Low.1 , (-1) , IFF (Low > High.1 , 1 , ((High - Low.1) - (High.1
- Low)) / &dvs))
&dvsv := IFF (Volume = 0 , 0.00001 , Volume)
MovingSum ((&mltp * Volume) , 20 , 0) / MovingSum (&dvsv , 20 , 0)
```

To recreate this indicator manually, click on the *edit* dropdow n menu and open the trader’s toolbox (or use CTRL+T) and click on the functions tab. Now click on the *new* button, and a *new function* dialogue w indow w ill open. In its text box, type in the code for the highlight bar. Ensure there are no extra spaces at the end of each line. When completed, click on the *verify* button. You may be presented w ith an *add inputs* popup message if there are variables in the code. If so, click the yes button, then enter a value in the *default value* column. If all is w ell, w hen you click on the function tab, the code you entered w ill convert to italic font. Now click the save button and type a name for the indicator.

Once this is completed, you can insert this indicator into your chart (Figure 11) by opening the *charting* dropdow n menu, selecting the *add to chart* command, then on the indicators tab. Find your named indicator, select it, then click on the *add* button. Repeat this procedure for the other indicators as w ell if you w ish.

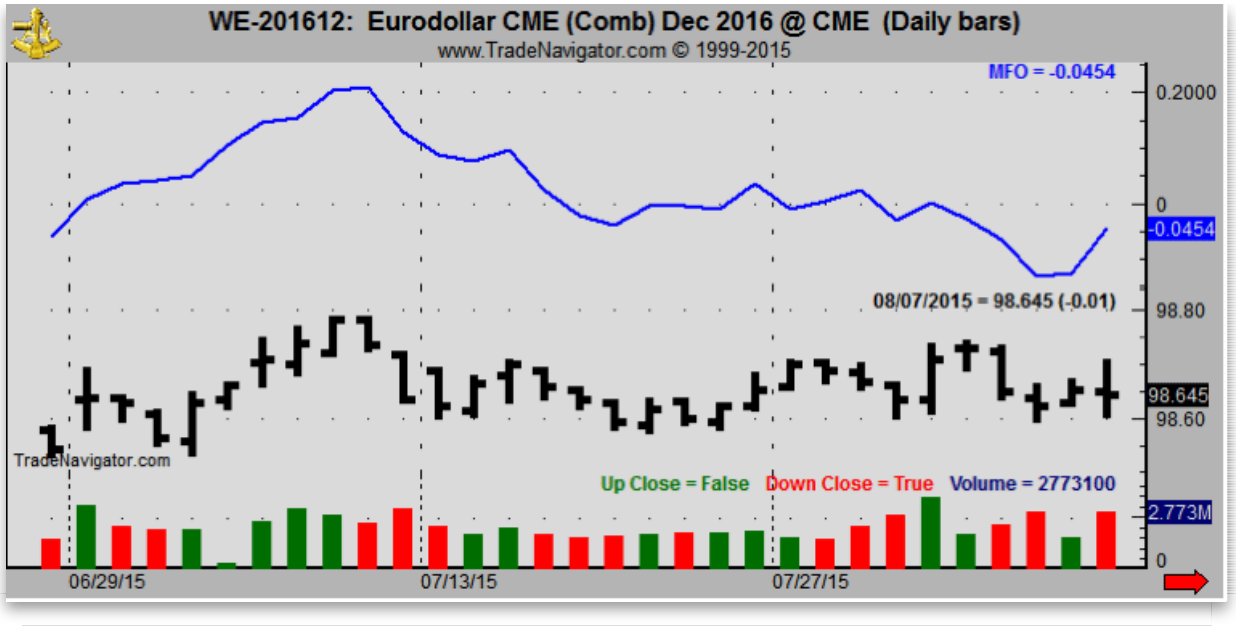


FIGURE 11: TRADENAVIGATOR. The MFO is shown on a chart of the eurodollar.

If you have any difficulties, call our technical support staff at (719) 884-0245 or click on the live chat tool located under Trade Navigator's help menu or near the top of our homepage.

—Customer service
Genesis Financial Technologies
www.TradeNavigator.com, 719 884-0245

BACK TO LIST

MICROSOFT EXCEL: OCTOBER 2015

The money flow oscillator that Vitali Apirine introduces in his article in this issue of the same name seems simple to calculate, less sensitive to price gaps, and seems to be able to keep you in trending markets. Certainly qualities I like to have in my indicator toolbox.

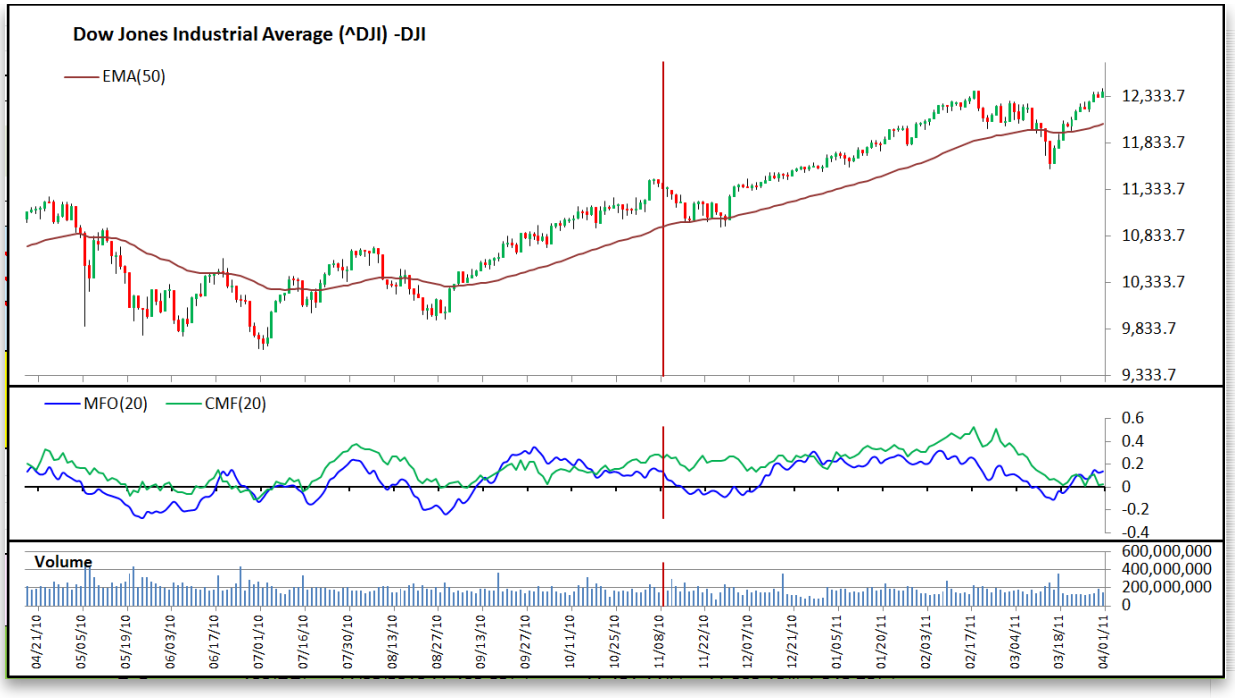


FIGURE 12: EXCEL. Here is an Excel-produced composite chart that approximates Figure 3 from Vitali Apirine's article in this issue (using a date range of 4/19/2010 to 4/1/2011).

Figure 12 show s an Excel-produced composite that approximates Figure 3 from Apirine's article in this issue (using a date range of 4/19/2010 to 4/1/2011). Similarly, Figure 13 approximates Figure 5 from Apirine's article in this issue (using a date range of 6/24/2013 to 2/6/2014).

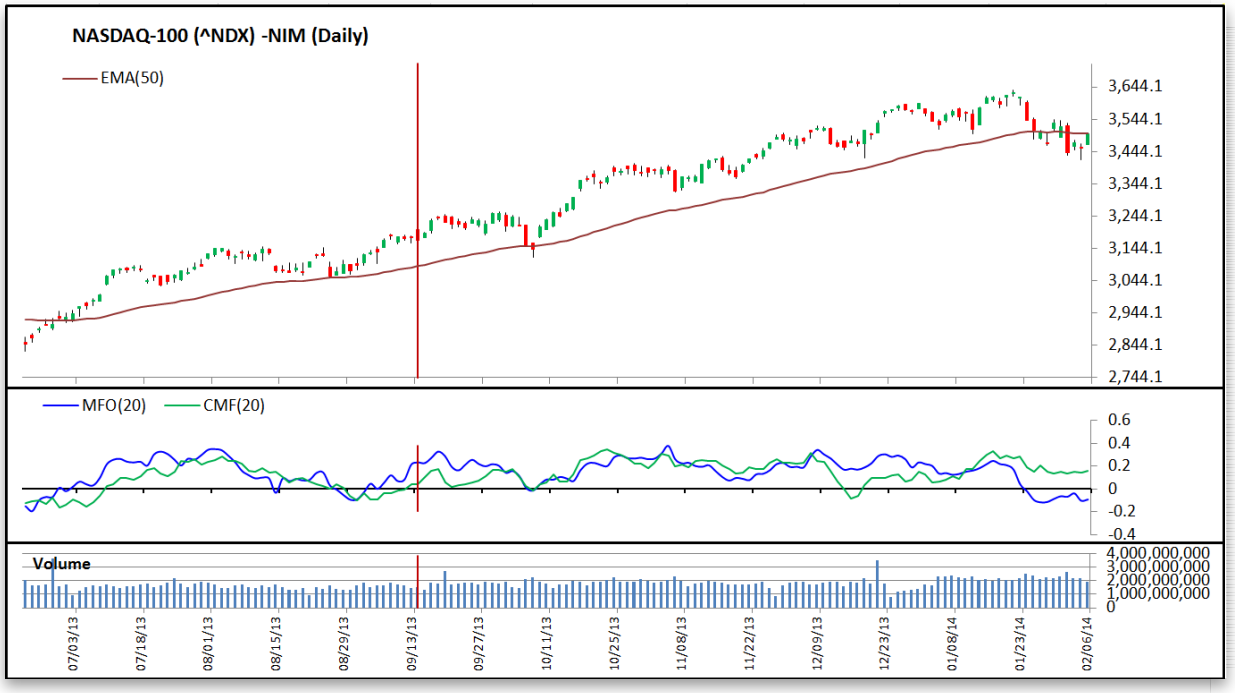


FIGURE 13 EXCEL. Here is an Excel-produced composite chart that approximates Figure 5 from Apirine's article in this issue (using a date range of 6/24/2013 to 2/6/2014).

The spreadsheet file for this Traders' Tip can be downloaded here. To successfully download it, follow these steps:

- Right-click on the Excel file link ("ApirineMoneyFLow Oscillator.xlsm"), then
- Select "save as" to place a copy of the spreadsheet file on your hard drive.

—Ron McAllister
Excel and VBA programmer
rpmac_xltt@sprynet.com

BACK TO LIST

Originally published in the October 2015 issue of
Technical Analysis of STOCKS & COMMODITIES magazine.

September 2015

For this month's Traders' Tips, the focus is John Ehlers' article in this issue, "Decyclers." Here, we present the September 2015 Traders' Tips code with possible implementations in various software.

The Traders' Tips section is provided to help the reader implement a selected technique from an article in this issue or another recent issue. The entries here are contributed by software developers or programmers for software that is capable of customization.

TRADESTATION: SEPTEMBER 2015

In "Decyclers" in this issue, author John Ehlers describes a method for constructing an oscillator that can help traders detect trend reversals with almost no lag. Ehlers calls this new oscillator a *decycler*. The author begins the process by isolating the high-frequency components present in the input data. He then subtracts these from the input data leaving only the lower-frequency components representing the trend.

For convenience, we are providing some EasyLanguage code for the simple decycler and decycler oscillator indicators based on Ehlers' concepts. In addition, we are providing code for a TradeStation function to calculate the decycler oscillator, and a strategy that demonstrates the function's usage.

Indicator: Simple Decycler

```
// TASC Sep 2015
// Decyclers - John F. Ehlers
// Simple Decyclers

inputs:
        HPPeriod( 125 ) ;

variables:
        alpha1( 0 ),
        HP( 0 ),
```

```
        Decycle( 0 ) ;

// High Pass Filter
alpha1 = ( Cosine ( 0.707 * 360 / HPPeriod ) +
Sine( 0.707 * 360 / HPPeriod ) - 1 )
/ Cosine( .707 * 360 / HPPeriod ) ;

HP = ( 1 - alpha1 / 2 ) * ( 1 - alpha1 / 2 ) *
( Close - 2 * Close[1] + Close[2] ) + 2 *
( 1 - alpha1 ) * HP[1] - ( 1 - Alpha1 ) *
( 1 - alpha1 ) * HP[2] ;

Decycle = Close - HP ;

Plot1( Decycle, "DeCycle" ) ;
Plot2( 1.005 * Decycle, "UpperBand" ) ;
Plot3( 0.995 * Decycle, "LowerBand" ) ;
```

Indicator: Decycler Oscillator

```
// TASC Sep 2015
// Decyclers - John F. Ehlers
// Decycler Oscillator

inputs:
        HPPeriod( 125 ),
        K( 1 ) ;

variables:
        alpha1( 0 ),
        alpha2( 0 ),
        HP( 0 ),
        Decycle( 0 ),
        DecycleOsc( 0 ) ;


alpha1 = ( Cosine ( 0.707 * 360 / HPPeriod ) +
Sine( 0.707 * 360 / HPPeriod ) - 1 )
/ Cosine( .707 * 360 / HPPeriod ) ;

HP = ( 1 - alpha1 / 2 ) * ( 1 - alpha1 / 2 ) *
( Close - 2 * Close[1] + Close[2] ) + 2 *
( 1 - alpha1 ) * HP[1] - ( 1 - Alpha1 ) *
( 1 - alpha1 ) * HP[2] ;

Decycle = Close - HP ;


alpha2 = ( Cosine( 0.707 * 360 / ( 0.5 * HPPeriod ) )
+ Sine( 0.707 * 360 / ( 0.5 * HPPeriod ) ) - 1) /
 Cosine( 0.707 * 360 / ( 0.5 * HPPeriod ) ) ;

DecycleOsc = ( 1 - alpha2 / 2 ) * ( 1 - alpha2 / 2 ) *
( Decycle - 2 * Decycle[1] + Decycle[2] ) + 2 *
( 1 - alpha2 ) * DecycleOsc[1] - ( 1 - alpha2 ) *
( 1 - alpha2 ) * DecycleOsc[2] ;
```

```
Plot1( 100 * K * DecycleOsc / Close, "DecycleOsc" ) ;
Plot2( 0, "ZeroLine" ) ;
```

Function: DecyclerOscillator

```
// TASC Sep 2015
// Decyclers - John F. Ehlers
// Decycler Oscillator Function

inputs:
        Price( NumericSeries ),
        HPPeriod( NumericSimple ),
        K( NumericSimple ) ;

variables:
        alpha1( 0 ),
        alpha2( 0 ),
        HP( 0 ),
        Decycle( 0 ),
        DecycleOsc( 0 ) ;

// High Pass Filter
alpha1 = ( Cosine ( 0.707 * 360 / HPPeriod ) +
Sine( 0.707 * 360 / HPPeriod ) - 1 ) /
Cosine( .707 * 360 / HPPeriod ) ;

HP = ( 1 - alpha1 / 2 ) * ( 1 - alpha1 / 2 ) *
( Price - 2 * Price[1] + Price[2] ) + 2 *
( 1 - alpha1 ) * HP[1] - ( 1 - Alpha1 ) *
( 1 - alpha1 ) * HP[2] ;

Decycle = Price - HP ;

alpha2 = ( Cosine( 0.707 * 360 / ( 0.5 * HPPeriod ) )
+ Sine( 0.707 * 360 / ( 0.5 * HPPeriod ) ) - 1) /
Cosine( 0.707 * 360 / ( 0.5 * HPPeriod ) ) ;

DecycleOsc = ( 1 - alpha2 / 2 ) * ( 1 - alpha2 / 2 ) *
( Decycle - 2 * Decycle[1] + Decycle[2] ) + 2 *
( 1 - alpha2 ) * DecycleOsc[1] - ( 1 - alpha2 ) *
( 1 - alpha2 ) * DecycleOsc[2] ;

DecyclerOscillator = 100 * K * DecycleOsc / Price ;
```

Strategy: Decycler Oscillator

```
// TASC Sep 2015
// Decyclers - John F. Ehlers
// Decycler Oscillator Strategy

inputs:
        Price( Close ),
        Fast_HPPeriod( 100 ),
        Fast_K( 1.2 ),
        Slow_HPPeriod( 125 ),
        Slow_K( 1 ) ;
```

```
variables:
        Fast_Val( 0 ),
        Slow_Val( 0 ) ;

Fast_Val =
DecyclerOscillator( Price, Fast_HPPeriod, Fast_K ) ;

Slow_Val =
DecyclerOscillator( Price, Slow_HPPeriod, Slow_K ) ;

if Fast_Val crosses over Slow_Val then
        Buy next bar at Market
else if Slow_Val crosses over Fast_Val then
        SellShort next bar at Market ;
```

To dow nload the EasyLanguage code, please visit our TradeStation and EasyLanguage support forum. The code for this article can be found here: http://w w w .tradestation.com/TASC-2015. The ELD filename is "TASC_SEP2015.ELD." For more information about EasyLanguage in general, please see http://w w w .tradestation.com/EL-FAQ.

A sample chart is show n in Figure 1.

FIGURE 1: TRADESTATION. Here is an example of the simple decycler, decycler oscillator indicators, and decycler oscillator strategy applied to a daily chart of the SPY.

This article is for informational purposes. No type of trading or investment recommendation, advice, or strategy is being made, given, or in any manner provided by TradeStation Securities or its affiliates.

—Doug McCrary
TradeStation Securities, Inc.
www.TradeStation.com

BACK TO LIST

METASTOCK: SEPTEMBER 2015

In “Decyclers” in this issue, author John Ehlers presents a simple decycler and his decycle oscillator. MetaStock formulas based on his article are show n here:

Simple Decycler:

```
ti:= 125;{time periods}
alpha1:= (Cos(.707*360/ti) + Sin(.707*360/ti) -1)/Cos(.707*360/ti);
HP:= (1-alpha1/2)*(1-alpha1/2)*(C - Ref(2*C,-1) + Ref(C,-2)) +
2*(1-alpha1)*PREV - (1-alpha1)*(1-alpha1)*Ref(PREV,-1);
decycle:= C - hp;
Decycle;
1.005*decycle;
0.995*decycle
```

Decycle Oscillator:

```
ti:= 125;{time periods}
K:= 1;   {scaler}
alpha1:= (Cos(.707*360/ti) + Sin(.707*360/ti) -1)/Cos(.707*360/ti);
HP:= (1-alpha1/2)*(1-alpha1/2)*(C - Ref(2*C,-1) + Ref(C,-2)) +
2*(1-alpha1)*PREV - (1-alpha1)*(1-alpha1)*Ref(PREV,-1);
decycle:= C - hp;
alpha2:= (Cos(.707*360/(.5*ti)) + Sin(.707*360/(.5*ti)) -1)/Cos(.707*360/(.5*ti));
DecycleOsc:= (1-alpha2/2)*(1-alpha2/2)*(decycle - Ref(2*decycle,-1) +
Ref(decycle,-2)) + 2*(1-alpha2)*PREV - (1-alpha2)*(1-alpha2)*Ref(PREV,-1);
100*k*DecycleOsc/C
```

—William Golson
MetaStock Technical Support
www.metastock.com

BACK TO LIST

---

eSIGNAL: SEPTEMBER 2015

For this month's Traders' Tip, we're providing two studies: SimpleDecycler.efs and DecyclerOscillator.efs, based on the article in this issue by John Ehlers, titled "Decyclers." In the article, Ehlers presents a method of identifying trends with virtually no lag, unlike with moving averages. The SimpleDecycler study is overlaid on the chart (see Figure 2), whereas the DecyclerOscillator is displayed as a subchart (Figure 3).

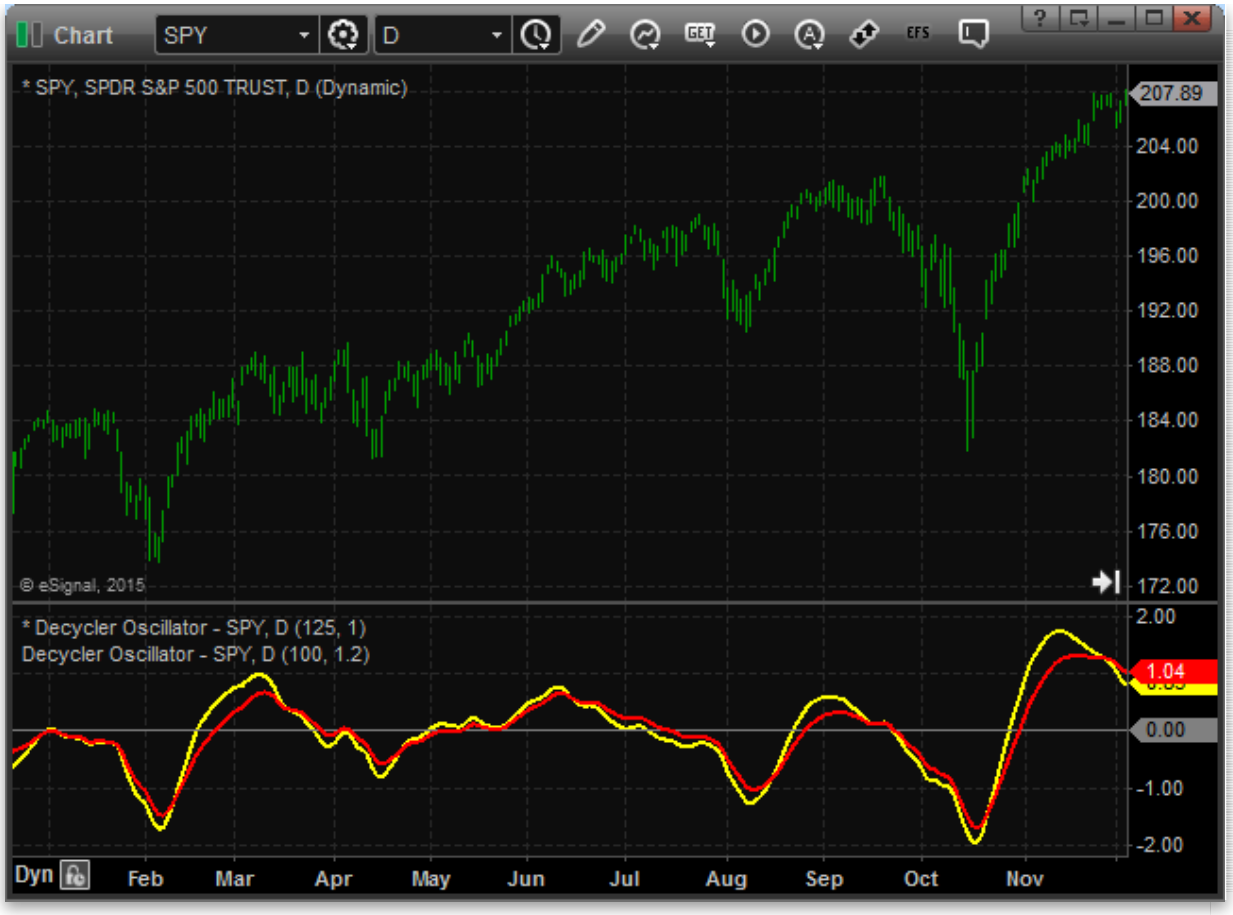


FIGURE 2: eSIGNAL. Here is an example of the SimpleDecycler study plotted on a daily chart of the SPY.

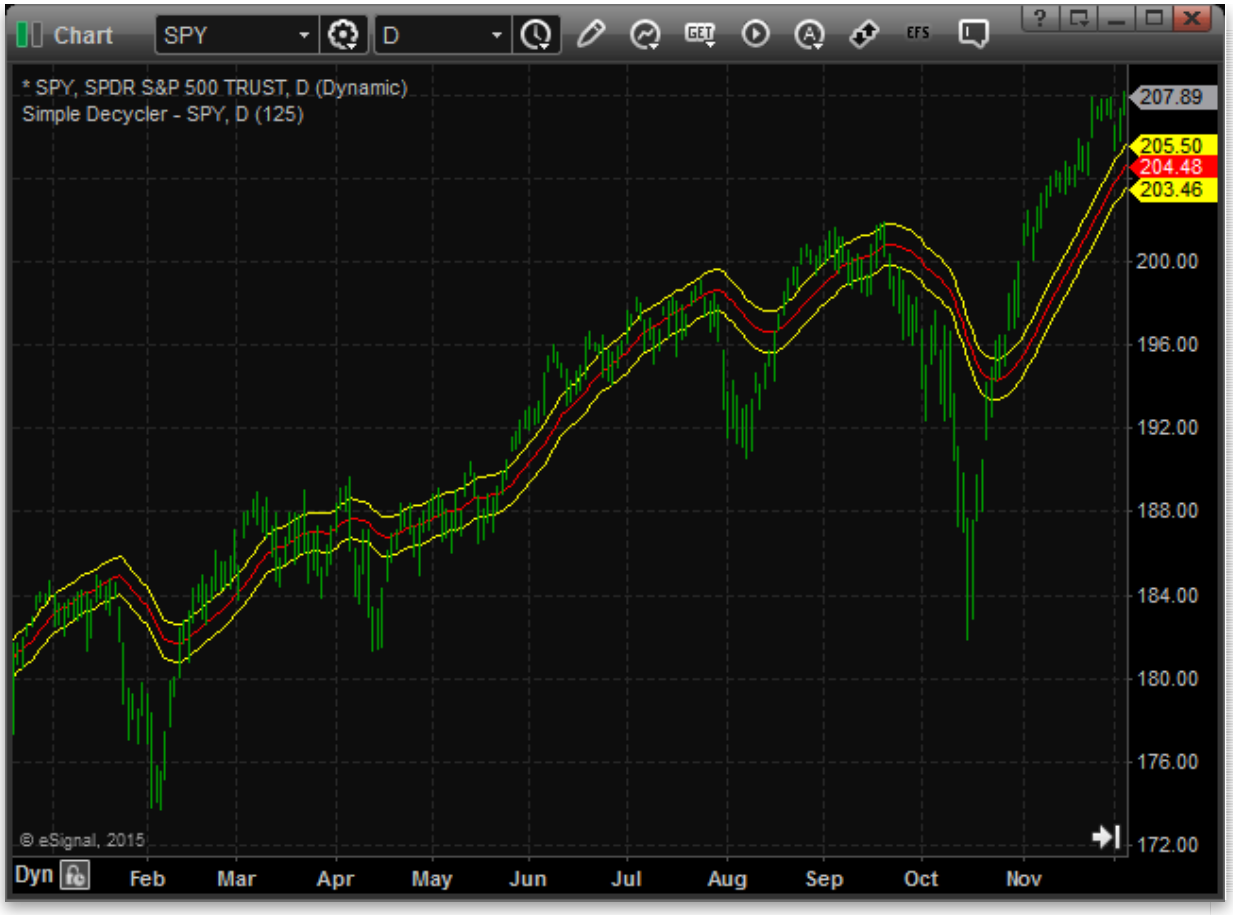


Figure 3: eSIGNAL. Here is an example of the DecyclerOscillator study plotted on a daily chart of the SPY.

The studies contain formula parameters that may be configured through the *edit chart* w indow (right-click on the chart and select "edit chart").

To discuss this study or dow nload a complete copy of the formula code, please visit the EFS Library Discussion Board forum under the *forums* link from the support menu at w w w .esignal.com or visit our EFS Know ledgeBase at http://w w w .esignal.com/support/kb/efs/. The eSignal formula script (EFS) is also available here:

SimpleDecycler.efs

```
/**********************************

Provided By:

    Interactive Data Corporation (Copyright © 2015)

    All rights reserved. This sample eSignal Formula Script (EFS)

    is for educational purposes only. Interactive Data Corporation

    reserves the right to modify and overwrite this EFS file with
```

```
    each new release.


Description:

    Decyclers by John Ehlers


Formula Parameters:                     Default:

    HP Period                           125


Version:            1.00  07/09/2015


Notes:

    The related article is copyrighted material. If you are not a subscriber

    of Stocks & Commodities, please visit www.traders.com.

**********************************/


var fpArray = new Array();


function preMain(){


    setStudyTitle("Simple Decycler");

    setPriceStudy(true);


    setCursorLabelName("UpperBand", 0);

    setCursorLabelName("Decycler", 1);

    setCursorLabelName("LowerBand", 2);


    setDefaultBarFgColor(Color.red, 1);


    var x = 0;
```

```
    fpArray[x] = new FunctionParameter("fpHPPeriod", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("HP Period");
        setLowerLimit(1);
        setDefault(125);
    }
}

var bInit = false;
var bVersion = null;

var xClose = null;
var xHP = null;

function main(fpHPPeriod){

    if (bVersion == null) bVersion = verify();
    if (bVersion == false) return;

    if (!bInit){

        xClose = close();
        xHP = efsInternal('Calc_HP', xClose, fpHPPeriod);

        bInit = true;
    };

    var nClose = xClose.getValue(0);
    var nHP = xHP.getValue(0);
```

```
    if (nHP == null || nClose == null)
        return;

    var nDecycle = nClose - nHP;

    return [nDecycle * 1.005, nDecycle, nDecycle * .995];
}

var nAlpha = null;

function Calc_HP(xSource, nPeriod){

    nAlpha = (Math.cos(.707*360 / nPeriod * Math.PI / 180) +
              Math.sin(.707*360 / nPeriod * Math.PI / 180) - 1) /
              Math.cos(.707*360 / nPeriod * Math.PI / 180);

    var nSource = xSource.getValue(0);
    var nSource1 = xSource.getValue(-1);
    var nSource2 = xSource.getValue(-2);

    if (nSource == null || nSource1 == null|| nSource2 == null)
        return;

    var nHP_1 = ref(-1);
    var nHP_2 = ref(-2);

    var nHP = Math.pow((1 - nAlpha / 2), 2) * (nSource - 2 * nSource1 + nSource2) +
              2 * (1 - nAlpha) * nHP_1 - Math.pow((1 - nAlpha), 2) * nHP_2;
```

```
    return nHP;

}


function verify(){


    var b = false;


    if (getBuildNumber() < 779){


        drawTextAbsolute(5, 35, "This study requires version 8.0 or later.",

            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,

            null, 13, "error");

        drawTextAbsolute(5, 20, "Click HERE to upgrade.@URL=http://www.esignal.com
/download/default.asp",

            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,

            null, 13, "upgrade");

        return b;

    }

    else

        b = true;


    return b;

}
```

DecyclerOscillator.efs

```
/*********************************

Provided By:

    Interactive Data Corporation (Copyright © 2015)

    All rights reserved. This sample eSignal Formula Script (EFS)
```

```
    is for educational purposes only. Interactive Data Corporation
    reserves the right to modify and overwrite this EFS file with
    each new release.

Description:
    Decyclers by John Ehlers

Formula Parameters:                     Default:
    HP Period                           125
    K                                   1

Version:            1.00  07/09/2015

Notes:
    The related article is copyrighted material. If you are not a subscriber
    of Stocks & Commodities, please visit www.traders.com.

**********************************/

var fpArray = new Array();

function preMain(){

    setStudyTitle("Decycler Oscillator");

    setCursorLabelName("ZeroLine", 0);
    setCursorLabelName("DecyclerOsc", 1);

    setDefaultBarFgColor(Color.grey, 0);
```

```
    setDefaultBarThickness(1, 0);
    setDefaultBarThickness(2, 1);

    setShowCursorLabel(false, 0);
    setShowCursorLabel(true, 1);

    var x = 0;

    fpArray[x] = new FunctionParameter("fpHPPeriod", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("HP Period");
        setLowerLimit(1);
        setDefault(125);
    }

    fpArray[x] = new FunctionParameter("fpK", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("K");
        setDefault(1);
    }
}

var bInit = false;
var bVersion = null;

var xClose = null;
var xHP = null;
var xDecycle = null;
var xDecycleOsc = null;
```

```
function main(fpHPPeriod, fpK){

    if (bVersion == null) bVersion = verify();
    if (bVersion == false) return;

    if (!bInit){

        xClose = close();
        xHP = efsInternal('Calc_HP', xClose, fpHPPeriod);
        xDecycle = efsInternal('Calc_Decycle', xClose, xHP);
        xDecycleOsc = efsInternal('Calc_HP', xDecycle, .5 * fpHPPeriod);

        bInit = true;
    };

    var nClose = xClose.getValue(0);
    var nDecycleOsc = xDecycleOsc.getValue(0);

    if (nDecycleOsc == null || nClose == null)
        return;

    var nReturnValue = 100 * fpK * nDecycleOsc / nClose;

    return [0, nReturnValue];
}

var nAlpha = null;

function Calc_HP(xSource, nPeriod){
```

```
    nAlpha = (Math.cos(.707*360 / nPeriod * Math.PI / 180) +
              Math.sin(.707*360 / nPeriod * Math.PI / 180) - 1) /
              Math.cos(.707*360 / nPeriod * Math.PI / 180);

    var nSource = xSource.getValue(0);
    var nSource1 = xSource.getValue(-1);
    var nSource2 = xSource.getValue(-2);

    if (nSource == null || nSource1 == null|| nSource2 == null)
        return;

    var nHP_1 = ref(-1);
    var nHP_2 = ref(-2);

    var nHP = Math.pow((1 - nAlpha / 2), 2) * (nSource - 2 * nSource1 + nSource2) +
              2 * (1 - nAlpha) * nHP_1 - Math.pow((1 - nAlpha), 2) * nHP_2;

    return nHP;
}

function Calc_Decycle(xSource, xHP){

    var nSource = xSource.getValue(0);
    var nHP = xHP.getValue(0);

    if (nHP == null || nSource == null)
        return;

    var nDecycle = nSource - nHP;
```

```
    return nDecycle;

}



function verify(){



    var b = false;



    if (getBuildNumber() < 779){



        drawTextAbsolute(5, 35, "This study requires version 8.0 or later.",

            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,

            null, 13, "error");

        drawTextAbsolute(5, 20, "Click HERE to upgrade.@URL=http://www.esignal.com
/download/default.asp",

            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,

            null, 13, "upgrade");

        return b;

    }

    else

        b = true;



    return b;

}
```

—Eric Lippert
eSignal, an Interactive Data company
800 779-6555, www.eSignal.com

BACK TO LIST

THINKORSWIM: SEPTEMBER 2015

In “Decyclers” in this issue, author John Ehlers presents the idea of *detrending* to build his decycler. In general, Ehlers discusses how cycle patterns go hand in hand with trends, so he built his decycler to go hand in hand with detrending studies.

We have built two new studies—the *simpler decycler* and *decycler oscillator*—for use in thinkorswim based on these concepts. The studies use our proprietary scripting language, thinkscript. We have made the loading process extremely easy; simply click on the following links: http://tos.mx/OZUx2C and http://tos.mx/HPPszc, and choose *save script to thinkorswim*, then choose to rename your studies to “SimplerDecycler” and “DecyclerOscillator,” in that order. You can adjust the parameters of these within the *edit studies* window to fine-tune your variables.

In the example chart in Figure 4, you see the SimplerDecycler plotted on top of the price graph and the DecyclerOscillator plotted in the subgraph.

FIGURE 4: THINKORSWIM. The SimplerDecycler is plotted on top of the price graph and the DecyclerOscillator is plotted in the subgraph.

—thinkorswim
A division of TD Ameritrade, Inc.
www.thinkorswim.com

BACK TO LIST

WEALTH-LAB: SEPTEMBER 2015

In his article in this issue, titled “Decyclers,” author John Ehlers shares two new decycler indicators whose main feature is the ability to identify trends with virtually zero lag.

We’ve developed a long-only trading system based on his concepts to demonstrate the timely response of the decycler oscillator to market action. It applies the idea of using a decycler oscillator pair with different parameters, as discussed in Ehlers’ article:

- **Enter long** when the yellow line crosses over the red line
- **Exit long** when the yellow line crosses under the red line

On the price pane, the decycler is plotted along with the hysteresis lines (+/- 0.5 percent). The green background indicates an uptrend in the market (the prices are above the upper hysteresis line) and the red background indicates a downtrend (the prices are below the lower hysteresis line). The bottom pane illustrates trend detection according to the pair of decycler oscillators. The yellow line has a period of 100 a *K* value of 1.2 and the red line has a period of 125 and a *K* value of 1.

As can be seen in Figure 5, robust application of the decycler bands in a trading system requires putting in some logic against whipsaw conditions such as from March 2015 to June 2015. The DecyclerOscillator seems to produce less whipsaws.

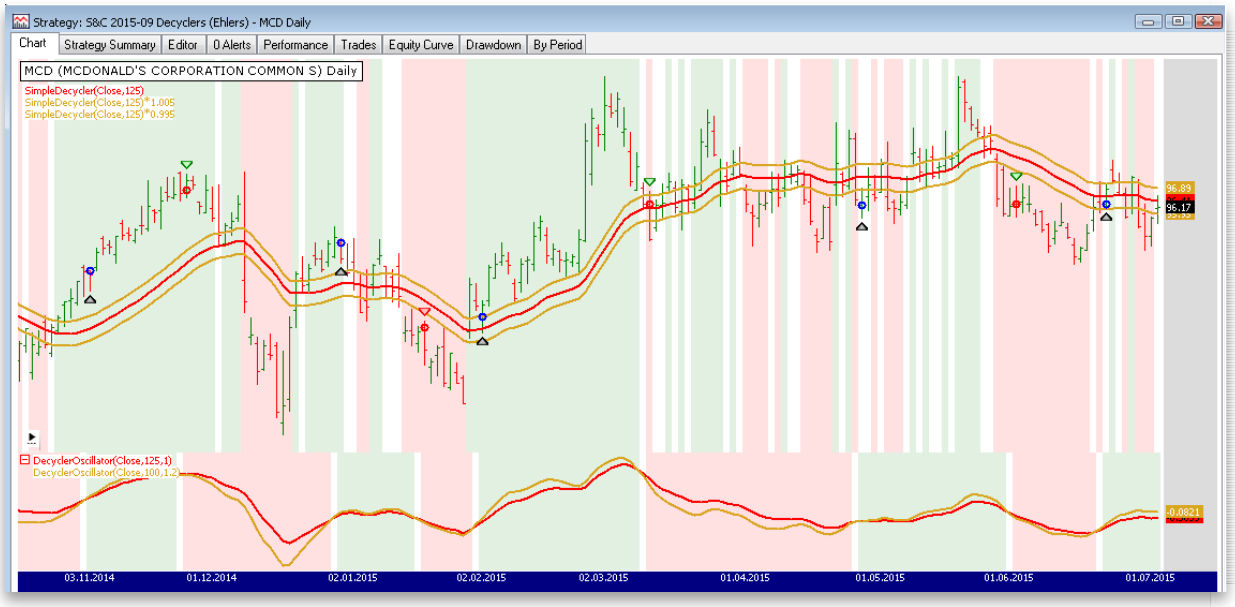


FIGURE 5: WEALTH-LAB. The price pane shows the decycler indicator with the hysteresis lines. The bottom pane plots a pair of decycler oscillators.

To execute the included trading system, Wealth-Lab users need to install (or update to) the latest version of the TASCIndicators library from the extensions section of our website if they haven’t already done so, and restart Wealth-Lab.

Wealth-Lab 6 code (C#):

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Drawing;
using WealthLab;
using WealthLab.Indicators;
using TASCIndicators;

namespace WealthLab.Strategies
{
        public class MyStrategy : WealthScript
        {
                protected override void Execute()
                {
                        // Decycler and oscillator
                        SimpleDecycler sd = SimpleDecycler.Series(Close,125);
            DecyclerOscillator do1 = DecyclerOscillator.Series(Close,125,1.0);
                        // Hysteresis band 0.5%
                        DecyclerOscillator do2 = DecyclerOscillator.Series(Close,
(int)(125 * 0.8),1.0 * 1.2);

                        HideVolume();
                        ChartPane paneDecyclerOsc = CreatePane(30,false,true);
                        PlotSeries(paneDecyclerOsc,do1,Color.Red,LineStyle.Solid,2);

PlotSeries(paneDecyclerOsc,do2,Color.Goldenrod,LineStyle.Solid,2);
                        PlotSeries(PricePane,sd,Color.Red,LineStyle.Solid,2);
                        PlotSeriesFillBand(PricePane, sd * 1.005, sd * 0.995,
Color.Goldenrod, Color.Transparent, LineStyle.Solid, 2);

                        for(int bar = GetTradingLoopStartBar(125); bar < Bars.Count;
bar++)
                        {
                                if ( Close[bar] > (sd[bar] * 1.005) )
                                        SetPaneBackgroundColor( PricePane, bar,
Color.FromArgb(30,Color.Green) );

                                else
                                        if ( Close[bar] < (sd[bar] * 0.995) )
                    SetPaneBackgroundColor( PricePane, bar,
Color.FromArgb(30,Color.Red) );

                                bool reverse2Uptrend = CrossOver( bar, do2, do1 );
                                bool reverse2Downtrend = CrossUnder( bar, do2, do1 );

                                if (IsLastPositionActive)
                                {
                                        if ( !reverse2Downtrend || reverse2Uptrend )
                                                SetPaneBackgroundColor(
paneDecyclerOsc, bar, Color.FromArgb(30,Color.Green) );

                                        if( reverse2Downtrend )
                                                SellAtMarket(bar+1, LastPosition);
                                }
                                else
                                {
                                        if ( reverse2Downtrend || !reverse2Uptrend )
                                                SetPaneBackgroundColor(
```

```
paneDecyclerOsc, bar, Color.FromArgb(30,Color.Red) );


                                        if( reverse2Uptrend )
                                                BuyAtMarket(bar+1);
                                }
                        }
                }
        }
}
```

—Eugene, Wealth-Lab team
MS123, LLC
www.wealth-lab.com

BACK TO LIST

AMIBROKER: SEPTEMBER 2015

In “Decyclers” in this issue, author John Ehlers presents an oscillator based on a second-order high-pass filtering technique. A ready-to-use AmiBroker formula is show n here. The formula is w ritten so it w orks w ith any AmiBroker version, but users of the most recent version may uncomment the part that uses the new ly introduced IIR function for speed. To use the oscillator, enter the code in the formula editor and press the *apply indicator* button. You can adjust the exponential average period and the RSI smoothing period using the *parameters* w indow .

A sample chart is show n in Figure 6.

FIGURE 6: AMIBROKER. Here is a daily chart of the SPY with decycle oscillators 125/1.0 and 100/1.2, which replicates one of the charts from John Ehlers' article in this issue.

AMIBROKER CODE LISTING.

```
SetBarsRequired( sbrAll );

function HighPass( input, alpha )
{
    output = input;

    for( i = 2; i < BarCount; i++ )
    {
      output[ i ] = ( 1 - alpha/2 ) ^ 2 *
                    ( input[ i ] - 2 * input[ i - 1 ] + input[ i -2 ] ) +
                   2 * ( 1 - alpha ) * output[ i - 1 ] -
                  ( 1 - alpha ) ^ 2 * output[ i - 2 ];
    }

    return output;
```

```
}

// newest version of AMiBroker (6.0) may use built-in IIR function instead
// - uncomment function below
/*
function HighPass( input, alpha )
{
   b0 = ( 1 - alpha/2 ) ^ 2;
   a1 = 2 * ( 1 - alpha );
   a2 = - ( 1 - alpha ) ^ 2;
   return IIR( input, b0, a1, -2 * b0, a2, b0 );
}
*/

function DecycleOsc( HPPeriod, K )
{
    alpha1 = 0;
    alpha2 = 0;
    HP = 0;
    Decycle = 0;

    // hipass filter
    PI = 3.1415926;

    angle1 = 0.707 * 2 * PI / HPPeriod;
    angle2 = 0.707 * 2 * PI / ( 0.5 * HPPeriod );

    alpha1 = ( cos( angle1 ) + sin( angle1 ) - 1 ) / cos( angle1 );
    alpha2 = ( cos( angle2 ) + sin( angle2 ) - 1 ) / cos( angle2 );

    HP = HighPass( Close, alpha1 );
    Decycle = Close - HP;
    return 100 * K * HighPass( Decycle, alpha2 );
}

Plot( DecycleOsc( 125, 1 )/Close, "DecycleOsc(125,1.0)",
      colorRed, styleThick );
Plot( DecycleOsc( 100, 1.2 )/Close, "DecycleOsc(100,1.2)",
      colorYellow, styleThick );
```

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

BACK TO LIST

NEUROSHELL TRADER: SEPTEMBER 2015

The quotient decycler and decycler oscillator indicators presented by John Ehlers in his article in this issue (“Decyclers”) can be easily implemented in NeuroShell Trader using NeuroShell Trader’s ability to call external dynamic linked libraries. Dynamic linked libraries may be w ritten in C, C++, Pow er Basic, or Delphi.

After moving the code given in Ehlers' article sidebar to your preferred compiler and creating a DLL, you can insert the resulting indicators as follows:

1. Select "New indicator" from the **Insert** menu.
2. Choose the **external program & library calls** category.
3. Select the appropriate **external DLL call** indicator.
4. Set up the parameters to match your DLL.
5. Select the *finished* button.

Similar filter and cycle-based strategies can also be created using indicators found in John Ehlers' **Cybernetic** and **MESA91** NeuroShell Trader Add-ons. Users of NeuroShell Trader can go to the Stocks & Commodities section of the NeuroShell Trader free technical support website to download a copy of this or any previous Traders' Tips.

A sample chart is shown in Figure 7.

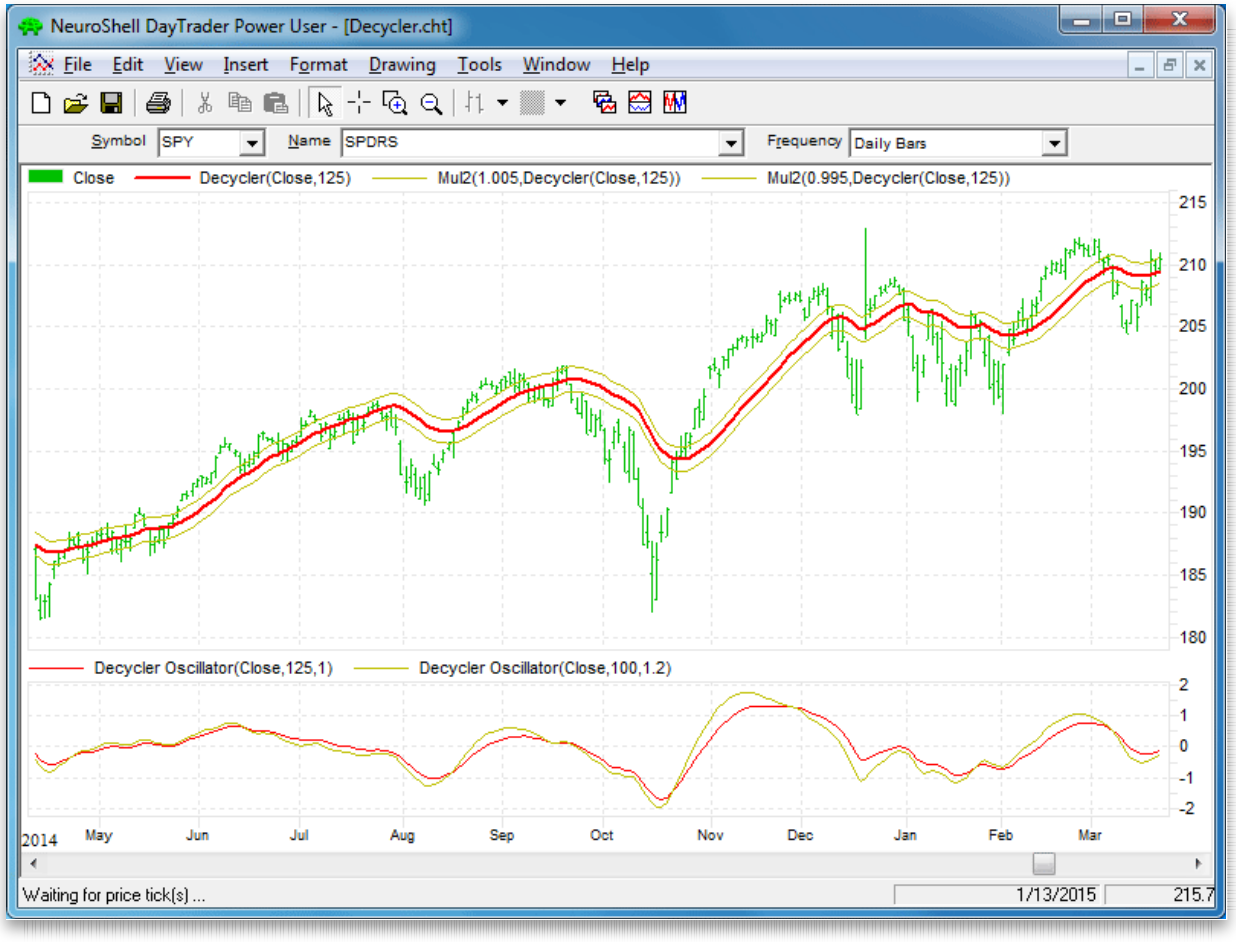


FIGURE 7: NEUROSHELL TRADER. This NeuroShell Trader chart displays the decycler and decyler oscillator indicators.

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

BACK TO LIST

TRADERSSTUDIO: SEPTEMBER 2015

The TradersStudio code based on John Ehlers' article in this issue, "Decyclers," can be found at: www.TradersEdgeSystems.com/traderstips.htm.

The following code files are provided in the download:

- Function EHLERS_DECYCLER: Computes the decycler values
- Indicator plot EHLERS_DECYCLER_IND: For plotting the decycler indicator on a chart
- Function EHLERS_DECYCLER_OSC: Computes the DecyclerOsc values
- Indicator plot EHLERS_DECYCLER_OSC_IND: For plotting the DecyclerOsc indicator on a chart.

Figure 8 shows the two indicators plotted on a chart of the S&P 500 futures contract (SP) using data from Pinnacle Data Corp.

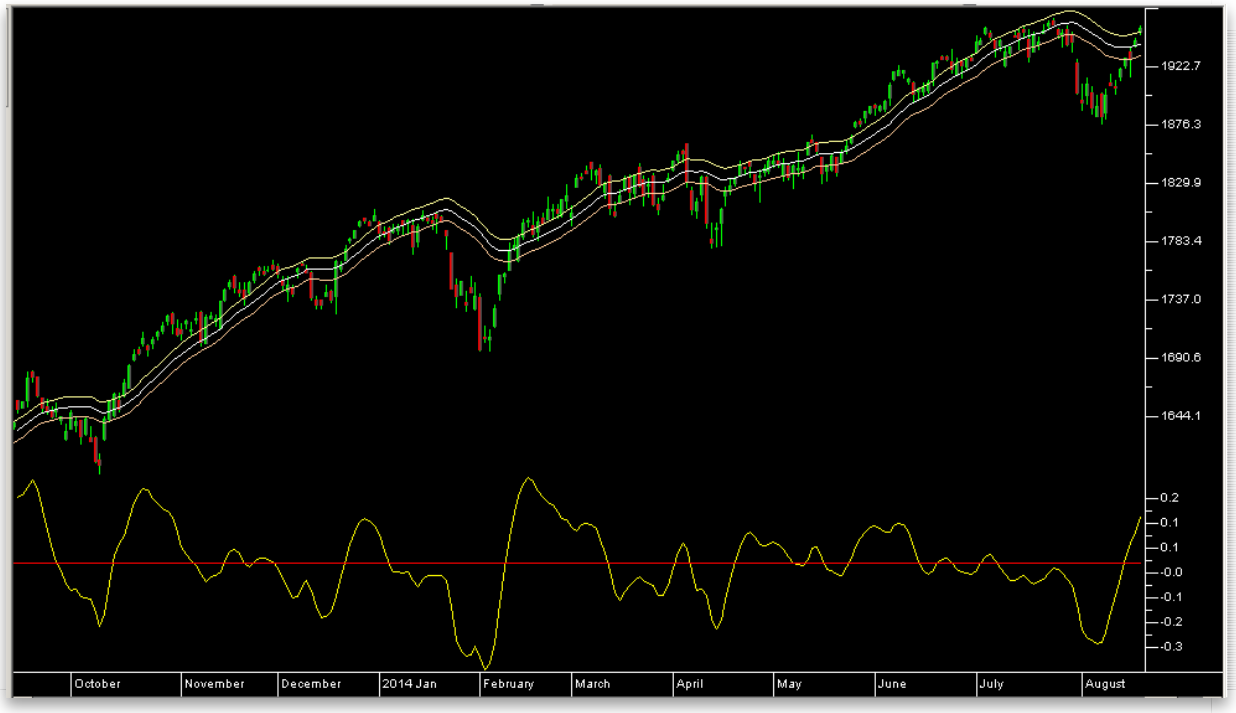


FIGURE 8: TRADERSSTUDIO. Here, the decycler indicators are shown on a chart of S&P 500 futures contract (SP).

The code is shown here as well as at the TradersEdgeSystems.com website.

```
'DECYCLERS

'Author: John F. Ehlers, TASC, Sept 2015

'Coded by: Richard Denning, 7/12/15
```

```
'www.TradersEdgeSystems.com



'HIGHPASS FILTER

Function EHLERS_DECYCLER(HPPeriod)

'HPPeriod=125

Dim alpha1

Dim HP As BarArray

If Cos(DegToRad(0.707*360/HPPeriod)) <> 0 Then

   alpha1 = (Cos(DegToRad(0.707*360/HPPeriod))+Sin(DegToRad(0.707*360/HPPeriod))-
1)/Cos(DegToRad(0.707*360/HPPeriod))

End If

HP = (1-alpha1/2)*(1-alpha1/2)*(C-2*C[1]+C[2])+2*(1-alpha1)*HP[1]-(1-alpha1)*
(1-alpha1)*HP[2]

EHLERS_DECYCLER = C - HP

End Function

'-------------------------------------------------------------------------------------
----------

Sub EHLERS_DECYCLER_IND(HPPeriod)

Dim Decycle As BarArray

Decycle = EHLERS_DECYCLER(HPPeriod)

Plot1(Decycle)

Plot2(1.005*Decycle)

Plot3(0.995*Decycle)

End Sub

'-------------------------------------------------------------------------------------
-----------

function EHLERS_DECYCLER_OSC(HPPeriod)

'HPPeriod=125,K=1

Dim Decycle As BarArray

Dim alpha2 As BarArray

Dim DecycleOsc As BarArray

Decycle = EHLERS_DECYCLER(HPPeriod)
```

```
alpha2=(Cos(DegToRad(0.707*360/(0.5*HPPeriod)))+Sin(DegToRad(0.707+360
/(0.5*HPPeriod))-1))/Cos(DegToRad(0.707-360/(0.5*HPPeriod)))

DecycleOsc=(1-alpha2/2)*(1-alpha2/2)*(Decycle-2*Decycle[1]+Decycle[2])+2*
(1-alpha2)*DecycleOsc[1]-(1-alpha2)*(1-alpha2)*DecycleOsc[2]

EHLERS_DECYCLER_OSC=DecycleOsc

End Function

'-----------------------------------------------------------------------------------------
-------------

sub EHLERS_DECYCLER_OSC_IND(HPPeriod,K)

Dim DecycleOsc As BarArray

DecycleOsc = EHLERS_DECYCLER_OSC(HPPeriod)

Plot1(100*K*DecycleOsc/C)

Plot2(0)

End Sub

'-----------------------------------------------------------------------------------------
-------------
```

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

NINJATRADER: SEPTEMBER 2015

The *simple decycler* and *decycler oscillator* indicators described by John Ehlers in his article in this issue, “Decyclers”, has been made available for dow nload at w w w .ninjatrader.com/SC/September2015SC.zip.

Once you have dow nloaded them, from w ithin the NinjaTrader Control Center w indow , select the menu File → Utilities → Import NinjaScript and select the dow nloaded file. This file is for NinjaTrader version 7.

You can review the indicators’ source code by selecting the menu Tools → Edit NinjaScript → Indicator from w ithin the NinjaTrader Control Center w indow and selecting either the SimpleDecycler or DecyclerOscillator file.

NinjaScript uses compiled DLLs that run native, not interpreted, to provide the highest possible performance.

A sample chart displaying the indicators is show n in Figure 9.

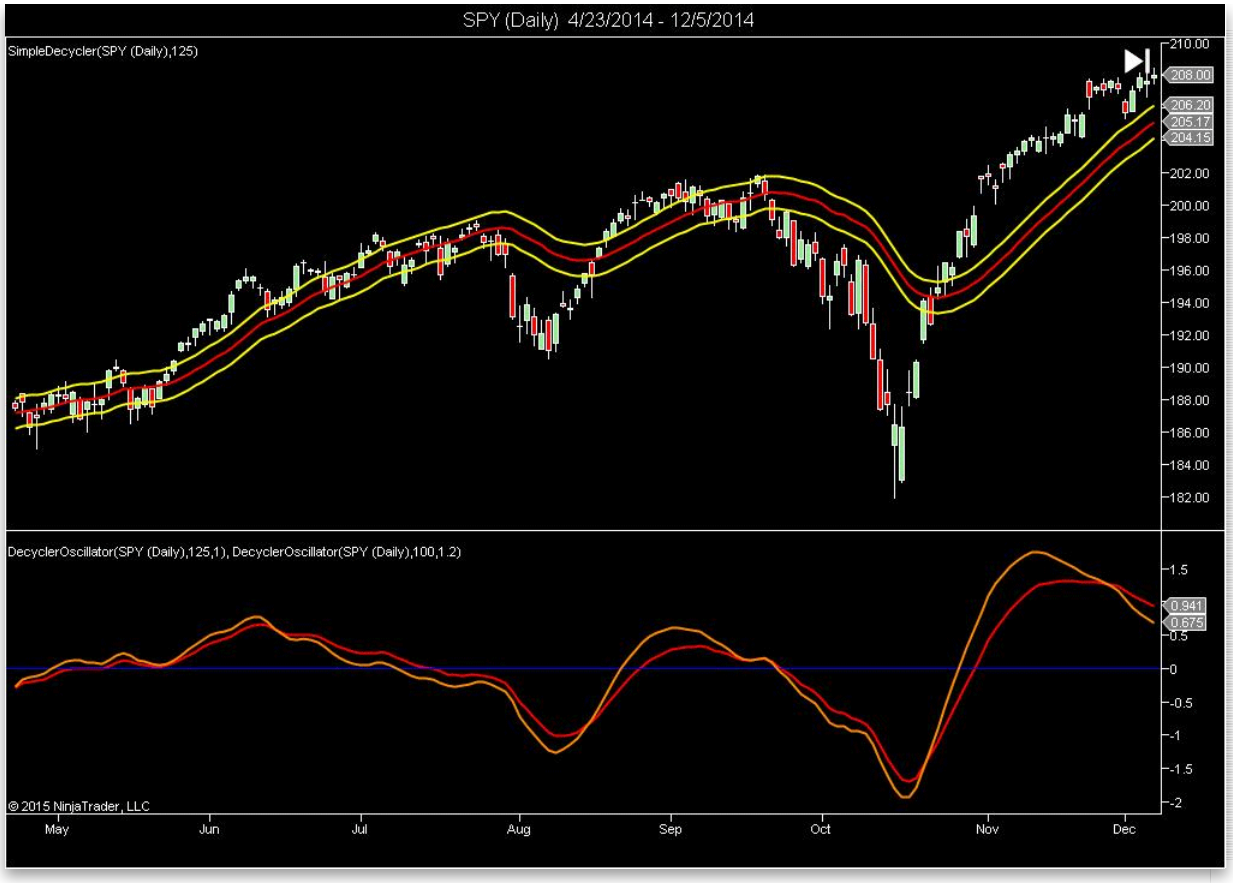


FIGURE 9: NINJATRADER. The decycler oscillator and simple decycler are displayed together on this sample SPY daily chart ended December 2014.

—Raymond Deux & Patrick Hodges
NinjaTrader, LLC
www.ninjatrader.com

BACK TO LIST

UPDATA: SEPTEMBER 2015

In "Decyclers" in this issue, author John Ehlers develops an oscillator that signals trend reversals with almost zero lag via digital signal processing techniques. A high-pass filter is subtracted from the input data and the high-frequency components are removed via cancellation of terms. Lower-frequency components are filtered from the output, so they are not canceled from the original data. Thus, the decycler displays them with close to zero lag.

The Updata code for both indicators is in the Updata library and may be downloaded by clicking the *custom* menu and *indicator library*. Those who cannot access the library due to a firewall may paste the code shown here into the Updata custom editor and save it.

```
'Decyclers
DISPLAYSTYLE 3LINES
INDICATORTYPE TOOL
COLOUR RGB(200,0,0)
COLOUR2 RGB(0,0,200)
COLOUR3 RGB(0,0,200)
PARAMETER "Period" #PERIOD=125
NAME "Decycler [" #PERIOD "]" ""
@ALPHA1=0
@HP=0
@DECYCLE=0
FOR #CURDATE=0 TO #LASTDATE
   'HIGHPASS FILTER
   @ALPHA1=1+(SIN(0.707*2*CONST_PI/#PERIOD)-1)/COS(0.707*2*CONST_PI/#PERIOD)
   @HP=(1-@ALPHA1/2)*(1-@ALPHA1/2)*(CLOSE-2*CLOSE(1)+CLOSE(2))+2*
(1-@ALPHA1)*HIST(@HP,1)-(1-@ALPHA1)*(1-@ALPHA1)*HIST(@HP,2)
   'DECYCLE THE DIFFERENCE BETWEEN INPUT AND HP DATA
   @DECYCLE=CLOSE-@HP
   @PLOT=@DECYCLE
   @PLOT2=1.005*@DECYCLE
   @PLOT3=0.995*@DECYCLE
NEXT

'DecyclerOscillator
DISPLAYSTYLE 2LINES
INDICATORTYPE CHART
COLOUR RGB(200,0,0)
COLOUR2 RGB(0,0,200)
COLOUR3 RGB(0,0,200)
PARAMETER "Period" #PERIOD=125
PARAMETER "K" @K=1
NAME "Decycler Osc. [" #PERIOD "] x [" @K "]" ""
@ALPHA1=0
@ALPHA2=0
@HP=0
@DECYCLE=0
@DECYCLEOSC=0
FOR #CURDATE=0 TO #LASTDATE
   'HIGHPASS FILTER
   @ALPHA1=1+(SIN(0.707*2*CONST_PI/#PERIOD)-1)/COS(0.707*2*CONST_PI/#PERIOD)
   @HP=(1-@ALPHA1/2)*(1-@ALPHA1/2)*(CLOSE-2*CLOSE(1)+CLOSE(2))+2*
(1-@ALPHA1)*HIST(@HP,1)-(1-@ALPHA1)*(1-@ALPHA1)*HIST(@HP,2)
   'DECYCLE THE DIFFERENCE BETWEEN INPUT AND HP DATA
   @DECYCLE=CLOSE-@HP
   'HIGHPASS FILTER OF DECYCLE TO CREATE THE OSC.
   @ALPHA2=1+(SIN(0.707*2*CONST_PI/(#PERIOD*0.5))-1)/COS(0.707*2*CONST_PI
/(#PERIOD*0.5))
   @DECYCLEOSC=(1-@ALPHA1/2)*(1-@ALPHA1/2)*(@DECYCLE-
2*HIST(@DECYCLE,1)+HIST(@DECYCLE,2))+2*(1-@ALPHA1)*HIST(@DECYCLEOSC,1)-(1-@ALPHA1)*
(1-@ALPHA1)*HIST(@DECYCLEOSC,2)
   @PLOT=100*@K*@DECYCLEOSC/CLOSE
   @PLOT2=0
NEXT
```

A sample chart is show n in Figure 10.

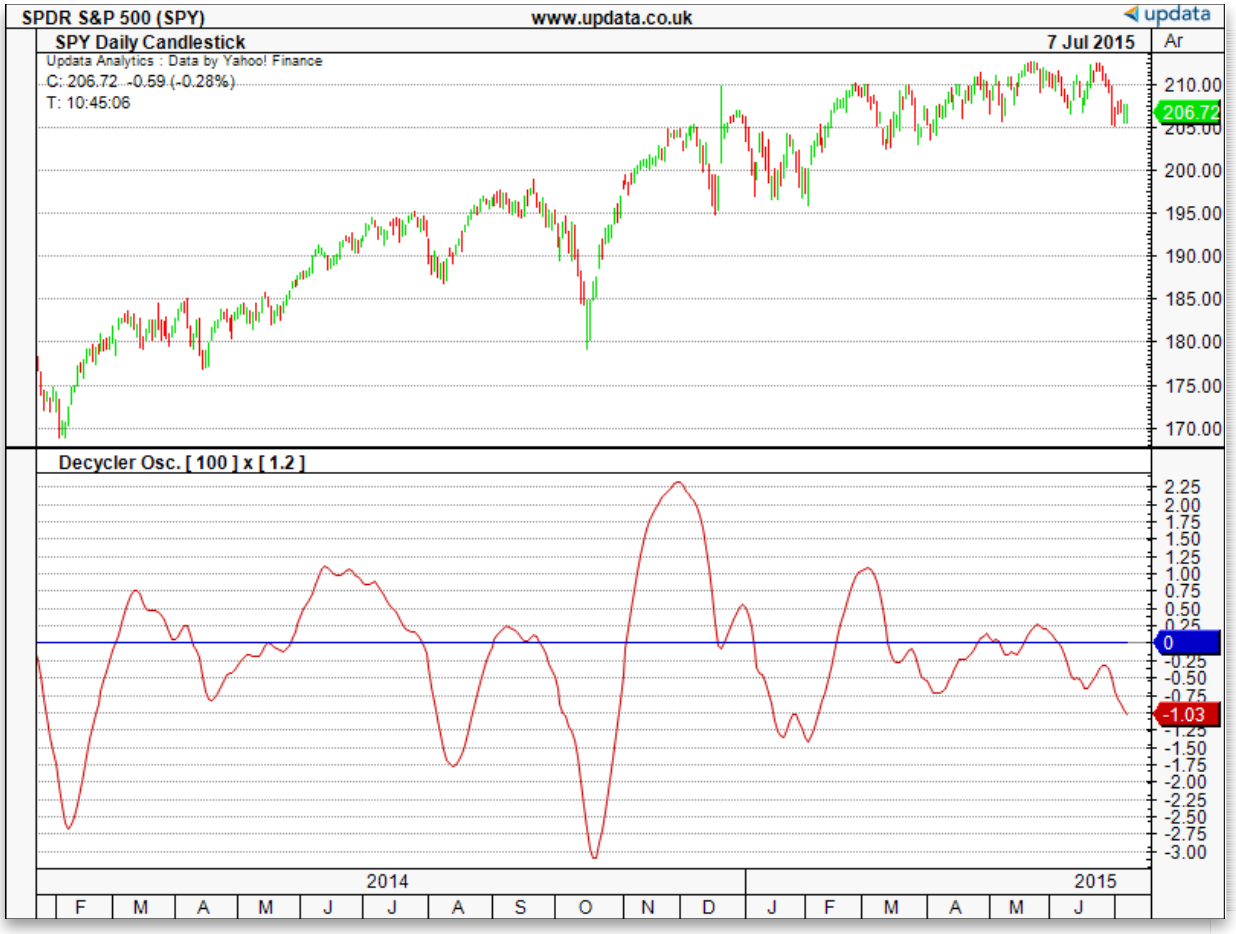


FIGURE 10: UPDATA. This chart shows the decycler oscillator [100][1.2] as applied to the SPY ETF of daily resolution.

—Updata support team
support@updata.co.uk, www.updata.co.uk

BACK TO LIST

MICROSOFT EXCEL: SEPTEMBER 2015

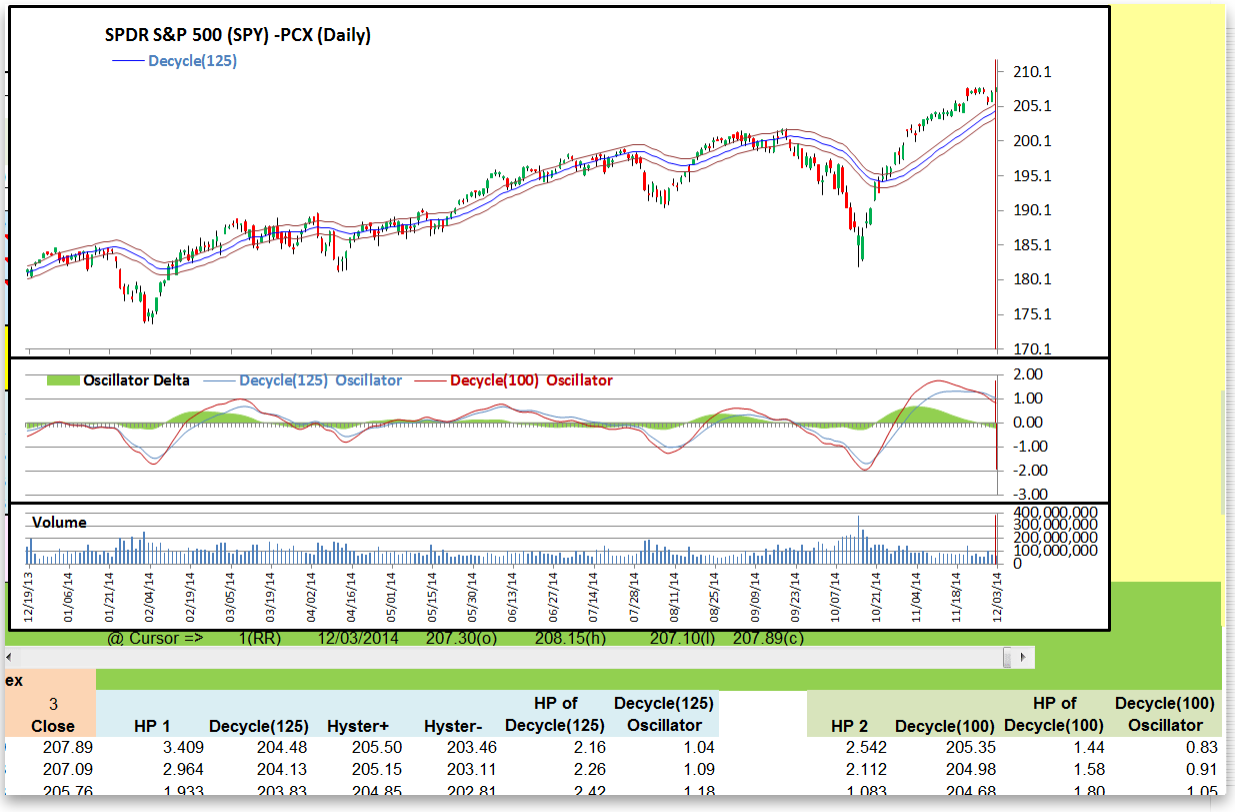
In his article in this issue, titled "Decyclers," author John Ehlers presents a simple, interesting new way to use the high-pass filter concept that he has discussed before in previous articles and contexts. The idea is to subtract the high-pass filter value from the current close to get a "decycled" indicator that follows the trend rather well by removing much of the distracting high-frequency chatter. Apply a second high-pass filter using half the period from the first filter to the decycled indicator and you get a smoother result with much-reduced lag.

Divide this second result by the close value and you have an oscillator whose turning points provide very good sensitivity to trend turning points of our data—perhaps even better (with a bit less lag) than the base decycled indicator.

Ehlers suggests calculating two such oscillators with differing high-pass periods and plotting them with appropriate scaling. Then use the crossover points to confirm trend changes.

As shown on my oscillator plot in Figure 11, I find that it is also interesting to plot the delta of the two

oscillators as a visual indication of the potential current strength of the indicated trend.



ex

3 Close	HP 1	Decycle(125)	Hyster+	Hyster-	HP of Decycle(125)	Decycle(125) Oscillator	HP 2	Decycle(100)	HP of Decycle(100)	Decycle(100) Oscillator
207.89	3.409	204.48	205.50	203.46	2.16	1.04	2.542	205.35	1.44	0.83
207.09	2.964	204.13	205.15	203.11	2.26	1.09	2.112	204.98	1.58	0.91
205.76	1.933	203.83	204.85	202.81	2.42	1.18	1.083	204.68	1.80	1.05

FIGURE 11: EXCEL, SPY. Here is an Excel-produced composite, which approximates Figures 1 & 2 from John Ehlers' article in this issue. The date range used here is 12/19/2013 to 12/03/2014.

Figure 12 show s the five user-adjustable control values for the decycler and oscillator computations.

Decyclers					
SPDR S&P 500 (SPY) -PCX (Daily)					
ClickToRefreshPrices					
User Charting Controls					
Data Control					
5652	Available Input Data Rows			Earliest:	01/29/1993
	Use Close Adjusted:	FALSE		Latest:	07/09/2015
841	Logically Available for Calc / Plot			$ Change:	2.300
Price Chart Windowing Controls				**Y-axis Range**	
149	Input Data Offset (Pan)		**Manual <= Set Chart**	**Max High+**	211.80
241	Data Points to Plot (Zoom)			**Min Low-**	170.10
	First:	Thu 12/19/2013		White space %:	5%
	Last:	Wed 12/03/2014		Allowance:	1.80
Decycler Controls		Half HP Period		Hysteresis Band Spread	0.5%
125	HPPeriod 1	62.5		Oscillator Scale Factor 1	1.0
100	HPPeriod 2	50.0		Oscillator Scale Factor 2	1.2
HPPeriod(125) Based Computational Factors					
	Alpha1	0.034921		Alpha2	0.068664
	(1-Alpha1/2)^2	0.965384		(1-Alpha2/2)^2	0.932515
	1-Alpha1	0.965079		1-Alpha2	0.931336
	(1-Alpha1)^2	0.931378		(1-Alpha2)^2	0.867386
Help! I would like to plot data					

FIGURE 12: EXCEL, USER CONTROLS. This shows the user controls: decycler periods, scale factors, and band spread.

In Ehlers' article sidebar, he presents formulas for the high-pass filter and oscillator. In the formulas, there are eight computational factors that remain constant once the high-pass period has been selected. I chose to compute these once and use the results in the indicator cell formulas. This approach saves computational time and physical space in memory and on disk.

Figure 13 show s the corresponding constants for the second oscillator computations. These can be found to the right of the charts.

HPPeriod(100) Based Computational Factors			
Alpha1	0.043464	Alpha2	0.085119
(1-Alpha1/2)^2	0.957008	(1-Alpha2/2)^2	0.916692
1-Alpha1	0.956536	1-Alpha2	0.914881
(1-Alpha1)^2	0.914961	(1-Alpha2)^2	0.837007

FIGURE 13: EXCEL, COMPUTATIONS. This shows the "constants" for the second oscillator computations.

The spreadsheet file for this Traders' Tip can be downloaded here. To successfully download it, follow these steps:

- Right-click on the Excel file link, then
- Select "save as" (or "save target as") to place a copy of the spreadsheet file on your hard drive.

—Ron McAllister
Excel and VBA programmer
rpmac_xltt@sprynet.com

BACK TO LIST

Originally published in the September 2015 issue of
Technical Analysis of STOCKS & COMMODITIES magazine.

November 2015

For this month's Traders' Tips, the focus is Vitali Apirine's article in this issue, "Average Percentage True Range." Code for MetaStock is already provided in Apirine's article in this issue. Here, we present the November 2015 Traders' Tips code with possible implementations in various software.

The Traders' Tips section is provided to help the reader implement a selected technique from an article in this issue or another recent issue. The entries here are contributed by software developers or programmers for software that is capable of customization.

eSIGNAL: NOVEMBER 2015

For this month's Traders' Tip, we've provided a study named APTR.efs based on the formula described in Vitali Apirine's article in this issue, "Average Percentage True Range." In the article, Apirine presents a study that displays the average true range (ATR) as a percentage instead of as points as in J. Welles Wilder Jr.'s original ATR study.

The eSignal study contains formula parameters that may be configured through the *edit chart* window (right-click on the chart and select "edit chart"). A sample chart is shown in Figure 1.

FIGURE 1: eSIGNAL. Here is an example of the study plotted on a daily chart of $RUT along with the classic ATR for comparison.

To discuss this study or dow nload a complete copy of the formula code, please visit the EFS Library Discussion Board forum under the *forums* link from the support menu at w w w .esignal.com or visit our EFS Know ledgeBase at http://w w w .esignal.com/support/kb/efs/. The eSignal formula script (EFS) is also available as a dow nloadable file here.

```
/**********************************
Provided By:
    Interactive Data Corporation (Copyright B© 2015)
    All rights reserved. This sample eSignal Formula Script (EFS)
    is for educational purposes only. Interactive Data Corporation
    reserves the right to modify and overwrite this EFS file with
    each new release.

Description:
    Average Percentage True Range by Vitali Apirine

Formula Parameters:                     Default:
Length                                  14

Version:            1.00  09/09/2015

Notes:
    The related article is copyrighted material. If you are not a subscriber
    of Stocks & Commodities, please visit www.traders.com.
```

```
**********************************/

var fpArray = new Array();

function preMain(){

    setStudyTitle("APTR");

    setCursorLabelName("APTR", 0);

    var x = 0;

    fpArray[x] = new FunctionParameter("fpLength", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("Length");
        setLowerLimit(1);
        setDefault(14);
    }
}

var bInit = false;
var bVersion = null;

var xATRS = null;
var xAPTR = null;

function main(fpLength){

    if (bVersion == null) bVersion = verify();
    if (bVersion == false) return;

    if (!bInit){

        xATRS = efsInternal('Calc_ATRS');
        xAPTR = smma(fpLength, xATRS);

        bInit = true;
    };

    nAPTR = xAPTR.getValue(0);

    if (nAPTR == null)
        return;

    return [nAPTR * 100];
}

var xHigh = null;
var xLow = null;
var xClose = null;

function Calc_ATRS(){

    if (getBarState() == BARSTATE_ALLBARS){
        xHigh = high();
        xLow = low();
        xClose = close();
    }
```

```
    var nHigh = xHigh.getValue(0);
    var nLow = xLow.getValue(0);
    var nPrevClose = xClose.getValue(-1);

    if (nHigh == null || nLow == null || nPrevClose == null)
        return;

    var nHL = nHigh - nLow;
    var nHC = Math.abs(nHigh - nPrevClose);
    var nLC = Math.abs(nLow - nPrevClose);

    var nMax = Math.max(nHL, nHC, nLC);

    if (nMax == nHC) {
        nMid = nPrevClose + nMax / 2;
    }

    if (nMax == nLC || nMax == nHL){
        nMid = nLow + nMax / 2;
    }

    nATRS = nMax / nMid;

    return [nATRS];
}

function verify(){
    var b = false;
    if (getBuildNumber() < 3742){

        drawTextAbsolute(5, 35, "This study requires version 12.1 or later.",
            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,
            null, 13, "error");
        drawTextAbsolute(5, 20, "Click HERE to upgrade.@URL=http://www.esignal.com
/download/default.asp",
            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,
            null, 13, "upgrade");
        return b;
    }
    else
        b = true;

    return b;
}
```

—Eric Lippert
eSignal, an Interactive Data company
800 779-6555, www.eSignal.com

BACK TO LIST

In his article in this issue, “Average Percentage True Range,” author Vitali Apirine discusses how to improve the legacy indicator *average true range* originally developed by J. Welles Wilder Jr. We have recreated Apirine’s average percentage true range using our proprietary scripting language, *thinkscript*. We have made the loading process extremely easy; simply click on the link http://tos.mx/w q5ymQ and choose *save script to thinkorswim*, then choose to rename your study “ATRP.” You can adjust the parameters of this study within the *edit studies* window to fine-tune your variables.

FIGURE 2: THINKORSWIM. In this example chart, you see the new ATRP study as well as the legacy ATR study below the chart of the Russell 2000 index.

In the example chart shown in Figure 2, you see the new ATRP study as well as the legacy ATR study below the chart of the Russell 2000 index. We have drawn the purple regression line to display that the ATRP actually has a different value while the shape is very similar to that of the traditional ATR. Please see Apirine’s article in this issue for more on this indicator.

—thinkorswim
A division of TD Ameritrade, Inc.
www.thinkorswim.com

BACK TO LIST

In this issue's article "Average Percentage True Range," author Vitali Apirine presents a normalization of the classic average true range (ATR) indicator.

For obvious reasons, the original ATR indicator should not be used for making comparisons across different markets. Others have also recognized the problem in using absolute price values by the ATR before; for example, see the May 2006 article by John Forman in the STOCKS & COMMODITIES archive, "Cross-Market Evaluations With Normalized Average True Range" and the accompanying Wealth-Lab Traders' Tip in that issue (http://technical.traders.com/content/backissuearchive.asp), and see "Normalized Volatility Indicator" by Rajesh Kayakkal in the August 2010 issue and the accompanying Wealth-Lab Traders' Tip in that issue.

As a solution to the problem of using absolute price values, Wealth-Lab has been including a version of a normalized ATR indicator called "ATRP" (ATR percentage). It's simply an ATR multiplied by 100 and divided by the close price. We have coded the APTR and compared it visually to Apirine's ATRP. As can be seen on the chart in Figure 3, w hile the tw o are not exactly equal, they are substantially close in their readings and dynamics.

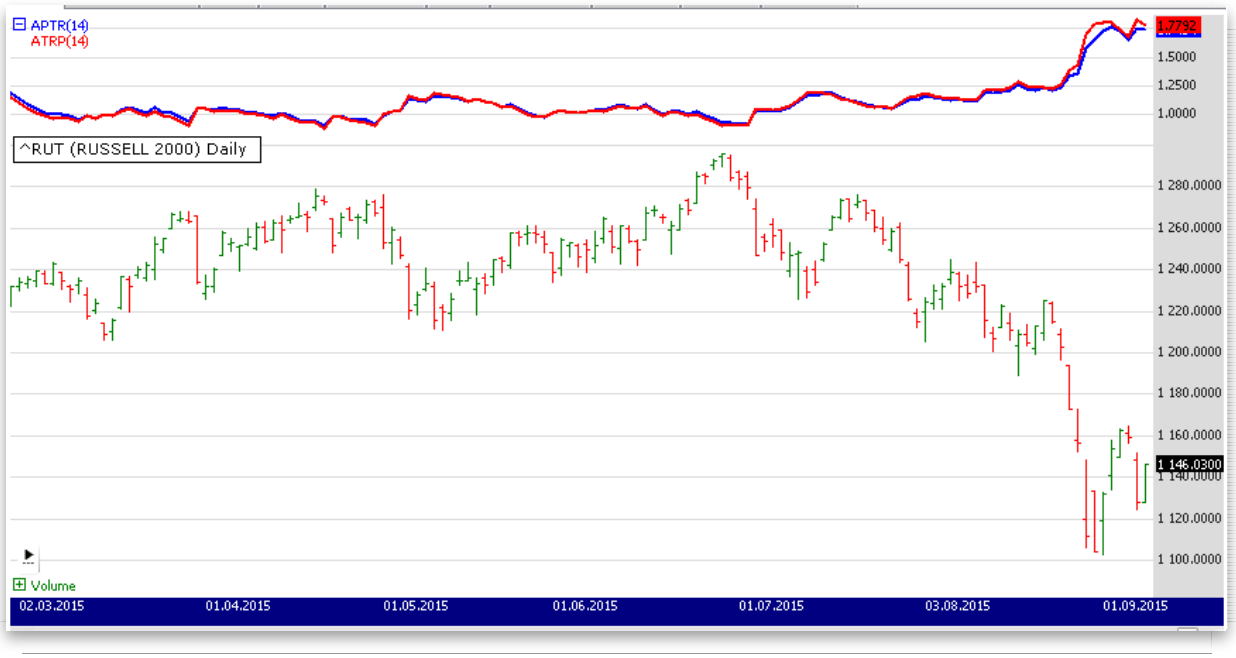


FIGURE 3: WEALTH-LAB. For the most part, both indicators' curves were inseparable during 2015. This sample chart shows the ATRP and APTR on the Russell 2000 index (^RUT daily data).

Usually, they reach extremes at about the same date; how ever, sometimes the red line (Wealth-Lab's ATRP) gets more responsive than the blue line (APTR). On such occasions, ATRP leads APTR for a short period. For example, Figure 4 compares the tw o indicators during the recent "Black Monday" phase on August 24, 2015.

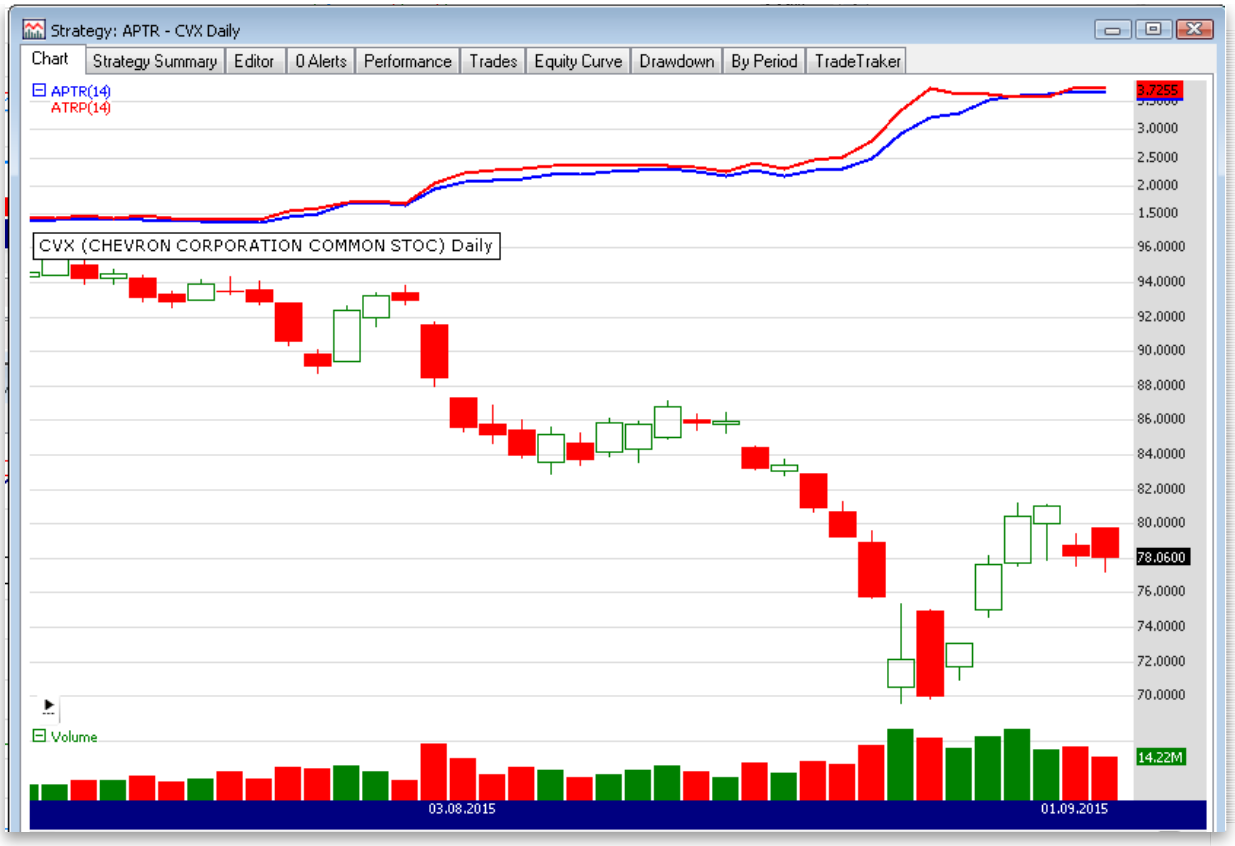


FIGURE 4 WEALTH-LAB. When volatility rapidly increases, ATRP becomes slightly more reactive than APTR, as shown here on a daily chart of Chevron (CVX).

After updating the TASCIndicators library to version 2015.10 or later, the APTR indicator can be found under the "TASC Magazine Indicators" group. You can plot it on a chart or use it as an entry or exit condition in a rule-based strategy without having to program a line of code yourself.

In conclusion, APTR is very similar to ATRP and can be used to measure volatility across different markets.

Wealth-Lab 6 strategy code (C#):

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Drawing;
using WealthLab;
using WealthLab.Indicators;
using TASCIndicators;

namespace WealthLab.Strategies
{
        public class MyStrategy : WealthScript
        {
                protected override void Execute()
                {
                        APTR aptr = APTR.Series(Bars,14);
                        ATRP atrp = ATRP.Series(Bars,14);
                        ChartPane aptrPane = CreatePane(30,true,true);
```

```
                        PlotSeries(aptrPane, aptr, Color.Blue, LineStyle.Solid, 2);
                        PlotSeries(aptrPane, atrp, Color.Red, LineStyle.Solid, 2);
                }
        }
}
```

—Eugene, Wealth-Lab team
MS123, LLC
www.wealth-lab.com

BACK TO LIST

AMIBROKER: NOVEMBER 2015

In “Average Percentage True Range” in this issue, author Vitali Apirine presents the ATR indicator that is based on percentage price movement. A ready-to-use formula for AmiBroker is show n here. To use the indicator, enter the code in the formula editor and press the *apply indicator* button. You can adjust the averaging period using the parameters w indow .

```
lh = High - Low;
pc = Ref( Close, -1 );
hc = abs( High - pc );
lc = abs( Low - pc );
MM = Max( Max( lh, hc ), lc );
atrs = IIF( MM == hc, hc / (  pc + ( hc / 2 ) ),
       IIF( MM == lc, lc / ( Low + ( lc / 2 ) ),
       IIF( MM == lh, lh / ( Low + ( lh / 2 ) ), 0 ) ) );

APTR = Wilders( atrs, Param("Period", 14, 1 ) ) * 100;

Plot( APTR, "APTR" + _PARAM_VALUES(), colorRed );
```

A sample chart is show n in Figure 5.

FIGURE 5: AMIBROKER. Here is a sample weekly chart of the Russell 2000 index, with a 14-bar APTR compared to the standard ATR. APTR gives a clearer signal of increased volatility thanks to using percentage price movement instead of absolute prices.

This and previously published AmiBroker Traders' Tips are available at http://w w w .amibroker.com/traders/.

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

BACK TO LIST

NEUROSHELL TRADER: NOVEMBER 2015

The average percent true range described by Vitali Apirine in his article in this issue, "Average Percentage

True Range," can be easily implemented w ith a few of NeuroShell Trader's 800+ indicators. Simply select *new indicator* from the *insert* menu and use the indicator w izard to set up the follow ing indicators:

```
HL% = Divide( Sub(High,Low), PriceMidpoint(High,Low) )
HC% = Divide( Abs(Sub(High,Lag(Close,1))), PriceMidPoint(High,Lag(Close,1),1) )
LC% = Divide( Abs(Sub(Low,Lag(Close,1))), PriceMidPoint(Low,Lag(Close,1), 1) )
APTR = Multiply2( ExpAvg(Max3(HL%,HC%,LC%), 14), 100)
```

Users of NeuroShell Trader can go to the Stocks & Commodities section of the NeuroShell Trader free technical support w ebsite to dow nload a copy of this or any previous Traders' Tips.

A sample chart of the APTR is show n in Figure 6.

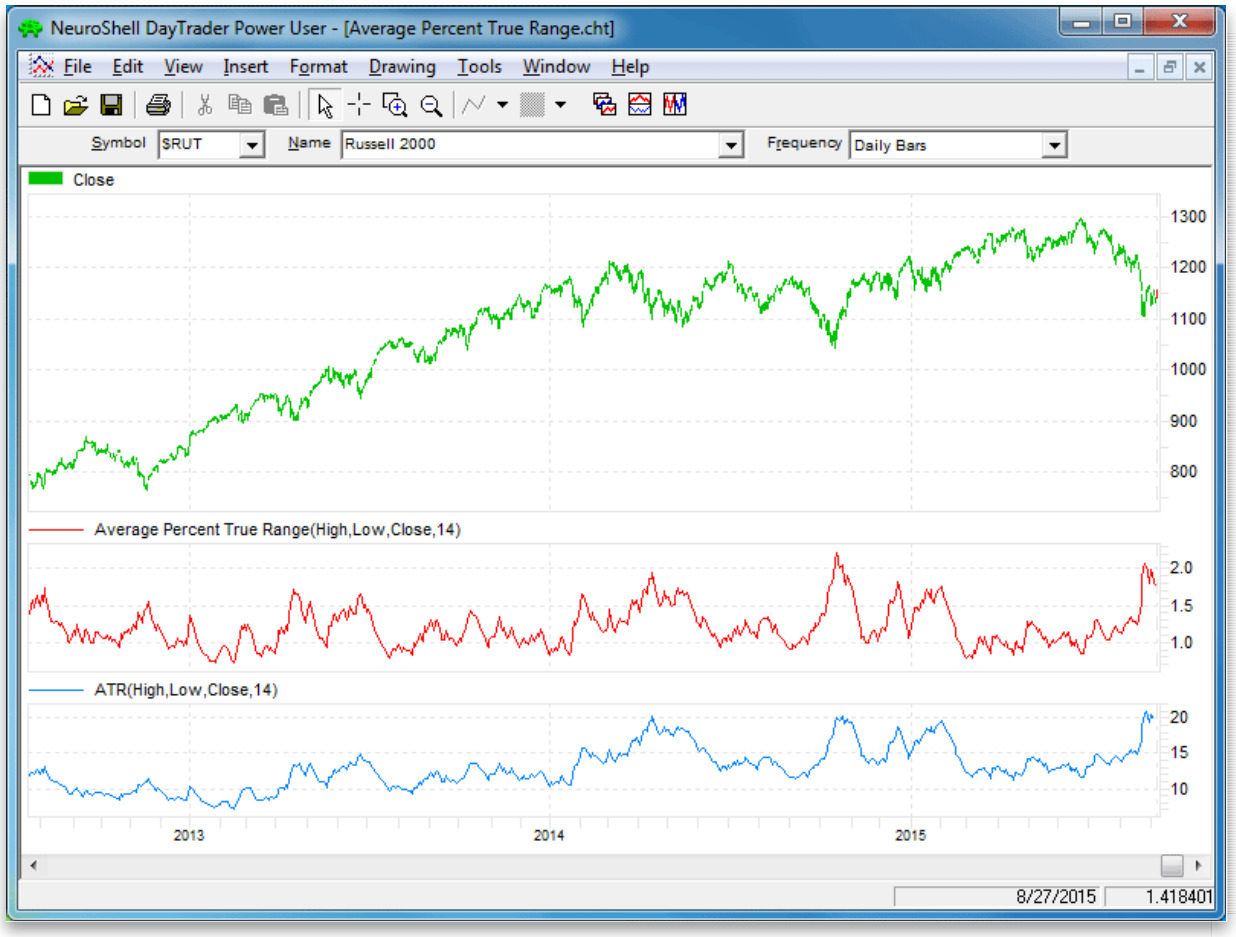


FIGURE 6: NEUROSHELL TRADER. This NeuroShell Trader chart displays the APTR on the Russell 2000 index.

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

BACK TO LIST

AIQ: NOVEMBER 2015

The AIQ code based on Vitali Apirine's article in this issue, "Average Percentage True Range," is provided at www.TradersEdgeSystems.com/traderstips.htm.

The code provided is used as an indicator (which I've called "PATR"). An example of the PATR is shown in Figure 7 on a chart of Apple Inc. (AAPL) compared to the same indicator on the S&P 500 index (SPX).

FIGURE 7: AIQ. Here is the percentage average true range (PATR) on a chart of AAPL in comparison to the same indicator plotted on the SPX index.

As mentioned, the code and EDS file can be downloaded from www.TradersEdgeSystems.com/traderstips.htm, and is shown below.

```
!AVERAGE PERCENTAGE TRUE RANGE
!Author: Vitali Apirine, TASC Nov 2015
!Coded by: Richard Denning 9/7/2015
!www.TradersEdgeSystems.com

WilderLen is 14.
Index is "SPX".
```

```
H is [high].
L is [low].
C is [close].
C1 is valresult(C,1).

LH is H - L.
HC is Abs(H - C1).
LC is abs(L - C1).

M is max(LH,HC).
MM is max(M,LC).

ATR1 is iff(MM=HC,HC,0).
MID1 is iff(ATR1>0,(valresult(C,1)+(HC/2)),0.00001).

ATR2 is iff(MM=LC and ATR1=0,LC,0).
MID2 is iff(ATR2>0,(L+(LC/2)),0.00001).

ATR3 is iff(MM=LH and ATR1=0 and ATR2=0,LH,0).
MID3 is iff(ATR3>0,(L+(LH/2)),0.00001).

ATRS is iff(ATR1>0,ATR1/MID1,iff(ATR2>0,ATR2/MID2,iff(ATR3>0,ATR3/MID3,0)))*100.

ExpLen is WilderLen*2-1.
APTR is expavg(ATRS,ExpLen). !PLOT

APTRidx is TickerUDF(Index,APTR). !PLOT

ShowValues if 1.
```

—Richard Denning
info@TradersEdgeSystems.com
for AIQ Systems

BACK TO LIST

TRADERSSTUDIO: NOVEMBER 2015

The TradersStudio code based on Vitali Apirine's article in this issue, "Average Percentage True Range," is provided at www.TradersEdgeSystems.com/traderstips.htm.

The following code files are provided in the download:

- **Function APTR**—Computes the average percentage true range values
- **Indicator plot APTR_IND**—For plotting the APTR indicator on a chart for the "mom" data series
- **Indicator plot APTRdata1_IND**—For plotting the APTR indicator on a chart for the first "child" data series.

Figure 8 shows the APTR indicator on a chart of Apple Inc. compared to the same indicator on the S&P 500 Index.

FIGURE 8: TRADERSSTUDIO. Here, the APTR indicator is displayed on a chart of Apple Inc. as well as on the S&P 500 index for comparison.

```
'AVERAGE PERCENTAGE TRUE RANGE
'Author: Vitali Apirine, TASC Nov 2015
'Coded by: Richard Denning 9/8/2015
'www TradersEdgeSystems com

Function APTR(WilderLen,pH As BarArray,pL As BarArray,pC As BarArray)
'WilderLen = 14
'Index = "SPX"
Dim LH As BarArray
Dim HC As BarArray
Dim LC As BarArray
Dim M As BarArray
Dim MM As BarArray
Dim ATR1 As BarArray
Dim MID1 As BarArray
Dim ATR2 As BarArray
Dim MID2 As BarArray
Dim ATR3 As BarArray
Dim MID3 As BarArray
Dim ATRS As BarArray
Dim ExpLen

LH = pH - pL
HC = Abs(pH - pC[1])
LC = Abs(pL - pC[1])

M = Max(LH,HC)
MM = Max(M,LC)

ATR1 = IFF(MM=HC,HC,0)
```

```
MID1 = IFF(ATR1>0,(pC[1]+(HC/2)),0.00001)

ATR2 = IFF(MM=LC And ATR1=0,LC,0)
MID2 = IFF(ATR2>0,(pL+(LC/2)),0.00001)

ATR3 = IFF(MM=LH And ATR1=0 And ATR2=0,LH,0)
MID3 = IFF(ATR3>0,(pL+(LH/2)),0.00001)

ATRS = IFF(ATR1>0,ATR1/MID1,IFF(ATR2>0,ATR2/MID2,IFF(ATR3>0,ATR3/MID3,0)))*100

ExpLen = WilderLen*2-1
APTR = XAverage(ATRS,ExpLen)

End Function
End Function
'-----------------------------------------------
'INDICATOR PLOT FOR THE MOM SERIES:
Sub APTR_IND(WilderLen)
plot1(APTR(WilderLen,H,L,C))
End Sub
'-----------------------------------------------
'INDICATOR PLOT FOR THE FIRST CHILD DATA SERIES:
Sub APTRdata1_IND(WilderLen)
plot1(APTR(WilderLen,H Of independent1,L Of independent1,C Of independent1))
End Sub
'-----------------------------------------------
```

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

NINJATRADER: NOVEMBER 2015

The *average percentage true range* (APTR) indicator, as presented in the article by Vitali Apirine in this issue, "Average Percentage True Range," has now been made available for download at www.ninjatrader.com/SC/November2015SC.zip.

Once you have it downloaded, from within the NinjaTrader Control Center window, select the menu File → Utilities → Import NinjaScript and select the downloaded file. This file is for NinjaTrader version 7.

You can review the indicator's source code by selecting the menu Tools → Edit NinjaScript → Indicator from within the NinjaTrader Control Center window and selecting the APTR file.

NinjaScript uses compiled DLLs that run native, not interpreted, to provide the highest possible performance.

A sample chart displaying the indicator is shown in Figure 9.

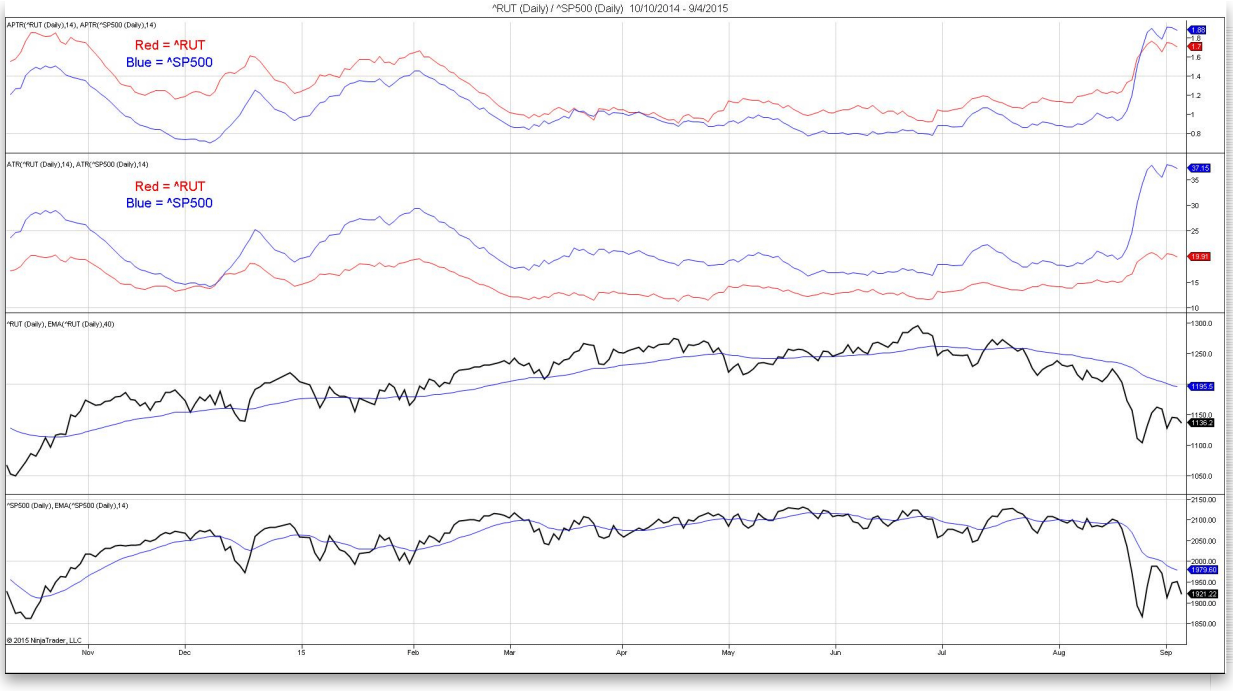


FIGURE 9: NINJATRADER. Here, the average percentage true range (APTR) is displayed with the average true range (ATR) for the ^RUT and ^SP500 (SPX) indexes.

—Raymond Deux & Zachary Gauld
NinjaTrader, LLC
www.ninjatrader.com

BACK TO LIST

UPDATA: NOVEMBER 2015

Our Traders' Tip for this month is based on the article in this issue by Vitali Apirine, "Average Percentage True Range."

In the article, Apirine modifies J. Welles Wilder's classic average true range calculation to use percentage price inputs instead of absolute prices in order to compare values w ith other instruments.

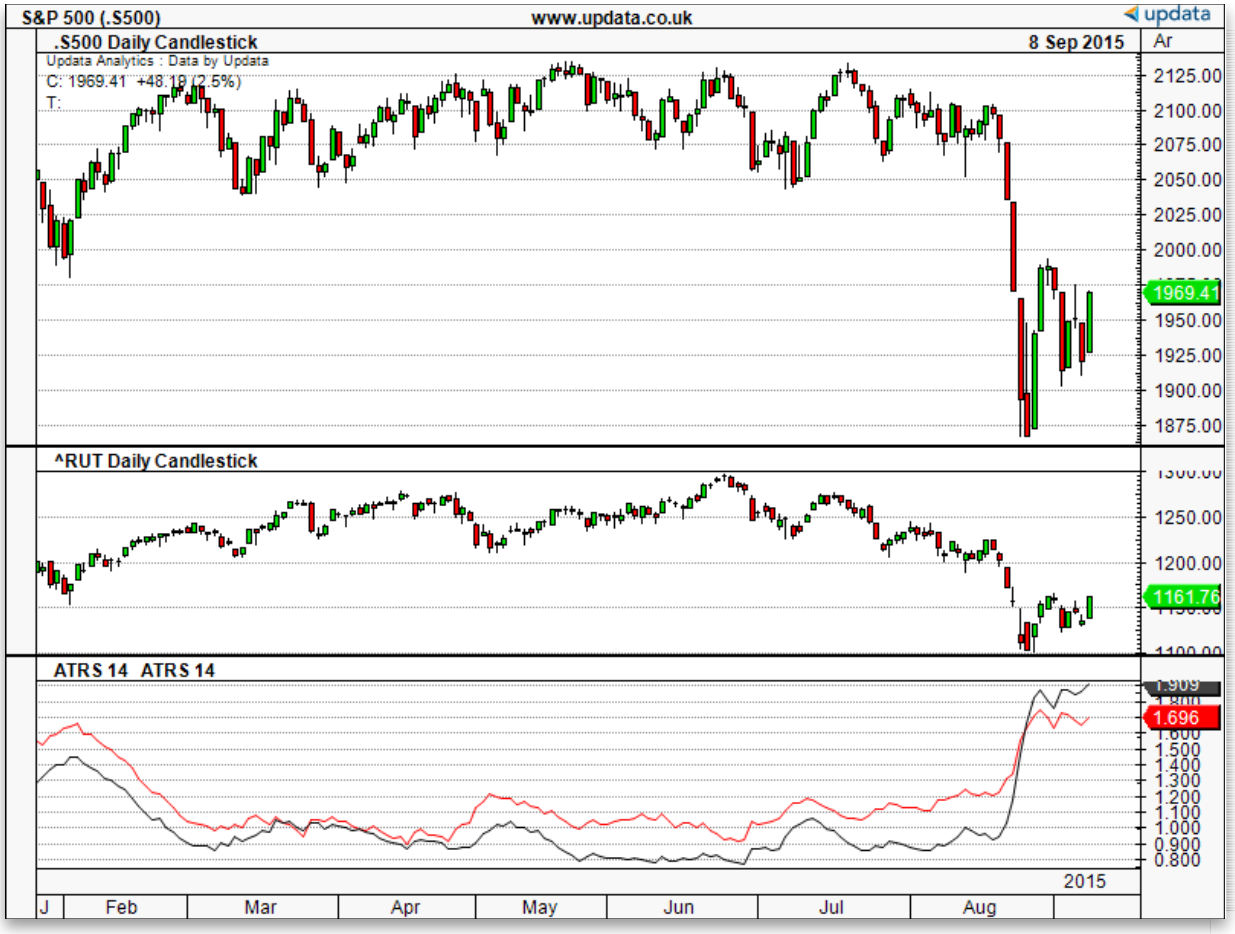


FIGURE 10: UPDATA. Here, the average percentage true range is applied to both the S&P 500 index and the Russell 2000 Index, plotted in the same window for comparison.

The Updata code based on this article is in the Updata library and may be downloaded by clicking the *custom* menu and then *indicator library*. Those who cannot access the library due to a firewall may paste the code shown here into the Updata custom editor and save it.

```
PARAMETER "Period" #PERIOD=14
NAME ATRS
@LH=0
@HC=0
@LC=0
@M=0
@MM=0
@ATR1=0
@MID1=0
@MID2=0
@MID3=0
@ATR2=0
@ATR3=0
@MID3=0
@ATRS=0
FOR #CURDATE=#PERIOD TO #LASTDATE
   @LH=HIGH-LOW
   @HC=ABS(HIGH-CLOSE(1))
```

```
    @LC=ABS(LOW-CLOSE(1))
    @M=MAX(@LH,@HC)
    @MM=MAX(@M,@LC)
    IF @MM=@HC
       @ATR1=@HC
    ELSE
       @ATR1=0
    ENDIF
    IF @ATR1>0
       @MID1=CLOSE(1)+@HC/2
    ELSE
       @MID1=0.00001
    ENDIF
    IF @MM=@LC
       @ATR2=@LC
    ELSE
       @ATR2=0
    ENDIF
    IF @ATR2>0
       @MID2=CLOSE(1)+@LC/2
    ELSE
       @MID2=0.00001
    ENDIF
    IF @MM=@LH
       @ATR3=@LH
    ELSE
       @ATR3=0
    ENDIF
    IF @ATR3>0
       @MID3=CLOSE(1)+@LH/2
    ELSE
       @MID3=0.00001
    ENDIF
    IF @ATR1>0
       @ATRS=@ATR1/@MID1
    ELSEIF @ATR2>0
       @ATRS=@ATR2/@MID2
    ELSEIF @ATR3>0
       @ATRS=@ATR3/@MID3
    ENDIF
    '6 DENOTES WILDER AVG
    @PLOT=100*SGNL(@ATRS,#PERIOD,6)
  NEXT
```

—Updata support team
support@updata.co.uk
www.updata.co.uk

BACK TO LIST

MICROSOFT EXCEL: NOVEMBER 2015

In “Average Percentage True Range” in this issue, author Vitali Apirine has provided us w ith an interesting w ay to translate volatility, as measured by the average true range, to a common scale. This allow s for a more direct comparison w hen trying to make a decision betw een several possible trades.

Figure 11 show s an Excel-produced composite that approximates Figure 2 from Apirine’s article (using a date

range of 7/14/1999 to 7/17/2000).

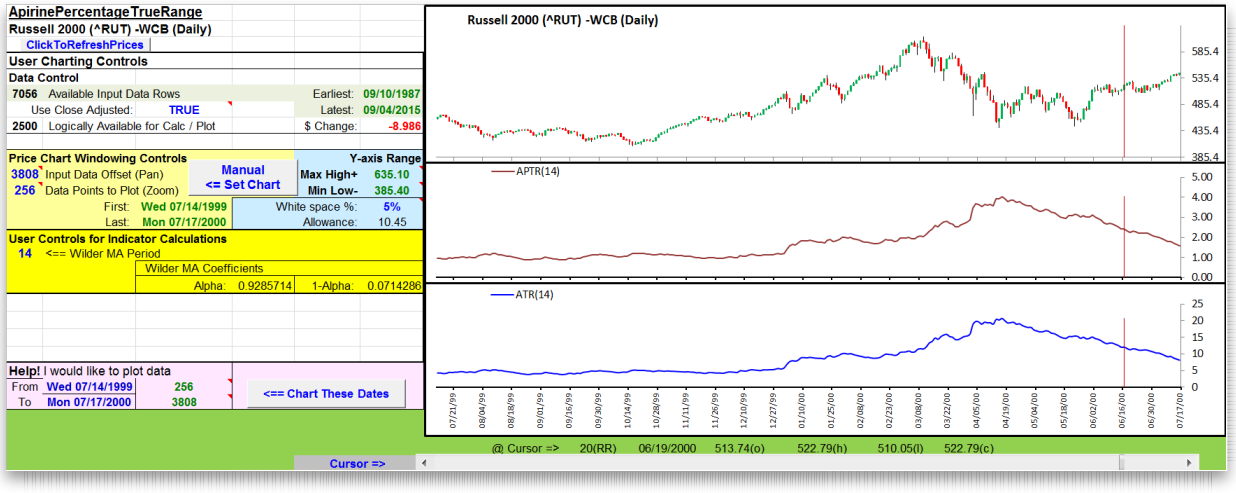


FIGURE 11: MICROSOFT EXCEL. Here is an Excel-produced composite that approximates Figure 2 from Vitali Apirine's article in this issue (using a date range of 7/14/1999 to 7/17/2000).

The spreadsheet file for this Traders' Tip can be downloaded here: ApirinePercentageTrueRange.xlsm. To successfully download it, follow these steps:

- Right-click on the Excel file link (ApirinePercentageTrueRange.xlsm), then
- Select "save as" (or "save target as") to place a copy of the spreadsheet file on your hard drive.

—Ron McAllister
Excel and VBA programmer
rpmac_xltt@sprynet.com

BACK TO LIST

Originally published in the November 2015 issue of
Technical Analysis of STOCKS & COMMODITIES magazine.

December 2015

For this month's Traders' Tips, the focus is Markos Katsanos's article in this issue, "Trading The Loonie." Here, we present the December 2015 Traders' Tips code with possible implementations in various software.

Code for MetaStock is already provided by Katsanos in the article, which S&C subscribers will find in the Subscriber Area of our website here.

The Traders' Tips section is provided to help the reader implement a selected technique from an article in this issue or another recent issue. The entries here are contributed by software developers or programmers for software that is capable of customization.

eSIGNAL: DECEMBER 2015

For this month's Traders' Tip, we've provided the studies BBDivergenceIndicator.efs and the BBDivergenceStrategy.efs based on the formulas described in Markos Katsanos's article in this issue, "Trading The Loonie."

In the article, the author presents a trading system that's based on the positive correlation between the currencies of major oil exporters and the price of oil. Katsanos focuses on the relationship of the Canadian dollar and the price of oil, and uses an indicator and strategy based on the divergence between them as indicated by a Bollinger Band study.

The eSignal studies contain formula parameters that may be configured through the *edit chart* window (right-click on the chart and select "edit chart"). A sample chart implementing the studies is shown in Figure 1.

FIGURE 1: eSIGNAL. Here is an example of the indicator and strategy plotted on a daily chart of CAD A0-FX.

To discuss these studies or dow nload a complete copy of the formula code, please visit the EFS Library discussion board forum under the *forums* link from the support menu at w w w .esignal.com or visit our EFS Know ledgeBase at http://w w w .esignal.com/support/kb/efs/. The eSignal formula scripts (EFS) are also show n below .

BBDivergenceIndicator.efs

```
/**********************************
Provided By:
    Interactive Data Corporation (Copyright B© 2015)
    All rights reserved. This sample eSignal Formula Script (EFS)
    is for educational purposes only. Interactive Data Corporation
    reserves the right to modify and overwrite this EFS file with
    each new release.

Description:
    Trading The Loonie by Markos Katsanos

Formula Parameters:                     Default:
Length BB Divergence                    20
Secondary Symbol                        CL #F
```

```
Version:            1.00  10/08/2015

Notes:
    The related article is copyrighted material. If you are not a subscriber
    of Stocks & Commodities, please visit www.traders.com.

**********************************/

var fpArray = new Array();

function preMain(){

    setStudyTitle("BBDivergenceStrategy");
    setIntervalsBackfill(true);

    var x = 0;

    fpArray[x] = new FunctionParameter("fpLength", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("Length BB Divergence");
        setLowerLimit(1);
        setDefault(20);
    }

    fpArray[x] = new FunctionParameter("fpSecondarySymbol", FunctionParameter.STRING);
    with(fpArray[x++]){
        setName("Secondary Symbol");
        setDefault('CL #F');
    }
}

var bInit = false;
var bVersion = null;

var xClose = null;
var xSecClose = null;

var xBol = null;
var xSecBol = null;

var xDiverg = null;

function main(fpLength, fpSecondarySymbol){

    if (bVersion == null) bVersion = verify();
    if (bVersion == false) return;

    if (!bInit){

        xClose = close();
        xSecClose = close(sym(fpSecondarySymbol));

        xBol = efsInternal('Calc_Bol', fpLength, xClose);
        xSecBol = efsInternal('Calc_Bol', fpLength, xSecClose);

        xDiverg = efsInternal('Calc_Div', xBol, xSecBol);

        bInit = true;
    }
```

```
    var nDiverg = xDiverg.getValue(0);

    if (nDiverg == null)
        return;

    return nDiverg;
}

var xBolSMA = null;
var xSTDDev = null;
var bInitInt = false;

function Calc_Bol(nLength, xSource){

    if (!bInitInt){
        xBolSMA = sma(nLength, xSource);
        xSTDDev = stdDev(nLength, xSource);

        bInitInt = true;
    }

    var nSource = xSource.getValue(0);
    var nBolSMA = xBolSMA.getValue(0);
    var nSTDDev = xSTDDev.getValue(0);

    if (nSource == null || nBolSMA == null || nSTDDev == null)
        return;

    var nBol = 1 + ((nSource - nBolSMA + 2 * nSTDDev) / (4 * nSTDDev + .0001));

    return nBol;
}

function Calc_Div(xBol1, xBol2){

    var nBol1 = xBol1.getValue(0);
    var nBol2 = xBol2.getValue(0);

    if (nBol1 == null || nBol2 == null)
        return;

    nDiverg = (nBol2 - nBol1) / nBol1 * 100 ;

    return nDiverg;
}

function verify(){
    var b = false;
    if (getBuildNumber() < 3742){

        drawTextAbsolute(5, 35, "This study requires version 12.1 or later.",
            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,
            null, 13, "error");
        drawTextAbsolute(5, 20, "Click HERE to upgrade.@URL=http://www.esignal.com
/download/default.asp",
            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,
            null, 13, "upgrade");
```

```
        return b;
    }
    else
        b = true;

    return b;
}
```

BBDivergenceStrategy.efs

```
/*********************************
Provided By:
    Interactive Data Corporation (Copyright B© 2015)
    All rights reserved. This sample eSignal Formula Script (EFS)
    is for educational purposes only. Interactive Data Corporation
    reserves the right to modify and overwrite this EFS file with
    each new release.

Description:
    Trading The Loonie by Markos Katsanos

Formula Parameters:                     Default:
Length BB Divergence                    20
Secondary Symbol                        CL #F
Long Trend Color                        green
Short Trend Color                       red

Version:            1.00  10/08/2015

Notes:
    The related article is copyrighted material. If you are not a subscriber
    of Stocks & Commodities, please visit www.traders.com.

*********************************/

var fpArray = new Array();

function preMain(){

    setStudyTitle("BBDivergenceStrategy");
    setPriceStudy(true);
    setIntervalsBackfill(true);

    var x = 0;

    fpArray[x] = new FunctionParameter("fpLength", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("Length BB Divergence");
        setLowerLimit(1);
        setDefault(20);
    }

    fpArray[x] = new FunctionParameter("fpSecondarySymbol", FunctionParameter.STRING);
    with(fpArray[x++]){
        setName("Secondary Symbol");
        setDefault('CL #F');
    }
```

```
    fpArray[x] = new FunctionParameter("fpLongColor", FunctionParameter.COLOR);
    with(fpArray[x++]){
        setName("Long Color");
        setDefault(Color.green);
    }

    fpArray[x] = new FunctionParameter("fpShortColor", FunctionParameter.COLOR);
    with(fpArray[x++]){
        setName("Short Color");
        setDefault(Color.red);
    }
}

var bInit = false;
var bVersion = null;

var xClose = null;
var xSecClose = null;

var xLow = null;
var xHigh = null;
var xOpen = null;

var xBol = null;
var xSecBol = null;

var xDiverg = null;

var xHHV = null;
var xHighHHV = null;
var xLLV = null;
var xLowLLV = null;
var xSecLLV = null;

var xROC = null;
var xSecROC = null;

var xCorrelEntry = null;
var xCorrelExit = null;

var xSecSMA = null;

var xMACD = null;
var xSignal = null;
var xStoch = null;

var nTagID = 0;

var nDefLotSize = null;

function main(fpLength, fpSecondarySymbol, fpLongColor, fpShortColor){

    if (bVersion == null) bVersion = verify();
    if (bVersion == false) return;

    if (!bInit){

        xClose = close();
        xSecClose = close(sym(fpSecondarySymbol));
```

```
        xLow = low();
        xHigh = high();
        xOpen = open();

        xBol = efsInternal('Calc_Bol', fpLength, xClose);
        xSecBol = efsInternal('Calc_Bol', fpLength, xSecClose);

        xDiverg = efsInternal('Calc_Div', xBol, xSecBol);

        xHHV = hhv(3, xDiverg);
        xHighHHV = hhv(15, xHigh);
        xLLV = llv(3, xDiverg);
        xLowLLV = llv(15, xLow);
        xSecLLV = llv(4, xSecClose);

        xROC = roc(2);
        xSecROC = roc(3, xSecClose);

        xCorrelEntry = efsInternal("Calc_Correlation", 20, xClose, xSecClose);
        xCorrelExit = efsInternal("Calc_Correlation", 60, xClose, xSecClose);

        xSecSMA = sma(40, xSecClose);

        xMACD = macd(12, 26, 9);
        xSignal = macdSignal(12, 26, 9);
        xStoch = stochD(30, 1, 3);

        bInit = true;
    }

    if (isLastBarOnChart()) return;

    var nDiverg = xDiverg.getValue(0);
    var nPrevDiverg = xDiverg.getValue(-1);

    var nClose = xClose.getValue(0);
    var nSecClose = xSecClose.getValue(0);

    var nHHV = xHHV.getValue(0);
    var nHighHHV = xHighHHV.getValue(-1);
    var nLLV = xLLV.getValue(0);
    var nLowLLV = xLowLLV.getValue(-1);
    var nSecLLV = xSecLLV.getValue(0);

    var nROC = xROC.getValue(0);
    var nSecROC = xSecROC.getValue(0);

    var nSecSMA = xSecSMA.getValue(0);
    var nPrev2SecSMA = xSecSMA.getValue(-2);

    var nCorrelEntry = xCorrelEntry.getValue(0);
    var nCorrelExit = xCorrelExit.getValue(0);

    var nStoch = xStoch.getValue(0);

    if (nDiverg == null || nPrevDiverg == null || nHHV == null || nHighHHV == null ||
        nLLV == null || nLowLLV == null || nSecLLV == null || nROC == null ||
nSecROC == null ||
        nSecSMA == null || nPrev2SecSMA == null || nCorrelEntry == null ||
nCorrelExit == null ||
```

```
        nStoch == null)
        return;

    nDefLotSize = Strategy.getDefaultLotSize();

    var nPrice = xOpen.getValue(1);

    if (!Strategy.isLong()){
        if ((nHHV > 20) && (nDiverg < nPrevDiverg) && (nROC > 0) && (nSecSMA >
nPrev2SecSMA) && (nCorrelEntry > -.4)){

            Strategy.doLong("Entry Long", Strategy.MARKET, Strategy.NEXTBAR,
Strategy.DEFAULT);
            drawShapeRelative(1, BelowBar1, Shape.UPTRIANGLE, null, fpLongColor,
Text.PRESET, nTagID++);
            drawTextRelative(1, BelowBar2, "Entry Long", fpLongColor, null,
Text.PRESET|Text.CENTER|Text.BOLD, null, null, nTagID++);
            drawTextRelative(1, BelowBar3, nDefLotSize + " @ " +
formatPriceNumber(nPrice), fpLongColor, null, Text.PRESET|Text.CENTER|Text.BOLD,
null, null, nTagID++);
        }
    }

    if (Strategy.isLong()){
        if ((crossAbove(xSignal, xMACD) && (nStoch > 85)) ||
            (nLLV < -20 && nSecROC < -3) ||
            (nClose < nLowLLV && nCorrelExit < -.4)){

            Strategy.doSell("Exit Long", Strategy.MARKET, Strategy.NEXTBAR,
Strategy.DEFAULT);
            drawShapeRelative(1, AboveBar1, Shape.SQUARE, null, fpShortColor,
Text.PRESET, nTagID++);
            drawTextRelative(1, AboveBar2, "Exit Long", fpShortColor, null,
Text.PRESET|Text.CENTER|Text.BOLD, null, null, nTagID++);
            drawTextRelative(1, AboveBar3, nDefLotSize + " @ " +
formatPriceNumber(nPrice), fpShortColor, null, Text.PRESET|Text.CENTER|Text.BOLD,
null, null, nTagID++);
        }
    }

    if (!Strategy.isShort()){
        if ((nLLV < -20) && (nDiverg > nPrevDiverg) && (nROC < 0) && (nSecSMA <
nPrev2SecSMA) && (nCorrelEntry > -.4)){

           Strategy.doShort("Entry Short", Strategy.MARKET, Strategy.NEXTBAR,
Strategy.DEFAULT);
           drawShapeRelative(1, AboveBar1, Shape.DOWNTRIANGLE, null, fpShortColor,
Text.PRESET, nTagID++);
           drawTextRelative(1, AboveBar2, "Entry Short", fpShortColor, null,
Text.PRESET|Text.CENTER|Text.BOLD, null, null, nTagID++);
           drawTextRelative(1, AboveBar3, nDefLotSize + " @ " +
formatPriceNumber(nPrice), fpShortColor, null, Text.PRESET|Text.CENTER|Text.BOLD,
null, null, nTagID++);
        }
    }

    if (Strategy.isShort()){
        if ((crossAbove(xMACD, xSignal) && (nStoch < 25) && nSecClose >= (1 + 4 /
100) * nSecLLV ) ||
            (nHHV > 20 && nSecROC > 4.5) ||
```

```
            (nClose > nHighHHV && nCorrelExit < -.4)){

            Strategy.doCover("Exit Short", Strategy.MARKET, Strategy.NEXTBAR,
Strategy.DEFAULT);
            drawShapeRelative(1, BelowBar1, Shape.SQUARE, null, fpLongColor,
Text.PRESET, nTagID++);
            drawTextRelative(1, BelowBar2, "Exit Short", fpLongColor, null,
Text.PRESET|Text.CENTER|Text.BOLD, null, null, nTagID++);
            drawTextRelative(1, BelowBar3, nDefLotSize + " @ " +
formatPriceNumber(nPrice), fpLongColor, null, Text.PRESET|Text.CENTER|Text.BOLD,
null, null, nTagID++);
        }
    }
}

function crossAbove(xSource1, xSource2){

    var bReturnValue = false;

    var nSource1 = xSource1.getValue(0);
    var nSource2 = xSource2.getValue(0);

    if (nSource1 > nSource2) {

        var i = -1;

        while (!bReturnValue){

            var nPrevSource1 = xSource1.getValue(i);
            var nPrevSource2 = xSource2.getValue(i);

            if (nPrevSource1 == null || nPrevSource2 == null)
                break;

            if (nPrevSource1 != nPrevSource2){
                if (nPrevSource2 > nPrevSource1)
                     bReturnValue = true;
                break;
            }
            i--;
        }
    }

    return bReturnValue;
}

var xBolSMA = null;
var xSTDDev = null;
var bInitInt = false;

function Calc_Bol(nLength, xSource){

    if (!bInitInt){
        xBolSMA = sma(nLength, xSource);
        xSTDDev = stdDev(nLength, xSource);

        bInitInt = true;
    }

    var nSource = xSource.getValue(0);
```

```
    var nBolSMA = xBolSMA.getValue(0);
    var nSTDDev = xSTDDev.getValue(0);

    if (nSource == null || nBolSMA == null || nSTDDev == null)
        return;

    var nBol = 1 + ((nSource - nBolSMA + 2 * nSTDDev) / (4 * nSTDDev + .0001));

    return nBol;
}

function Calc_Div(xBol1, xBol2){

    var nBol1 = xBol1.getValue(0);
    var nBol2 = xBol2.getValue(0);

    if (nBol1 == null || nBol2 == null)
        return;

    nDiverg = (nBol2 - nBol1) / nBol1 * 100 ;

    return nDiverg;
}

var xCorSMA1 = null;
var xCorSMA2 = null;

function Calc_Correlation(nLength, xSource1, xSource2){

    if (getBarState() == BARSTATE_ALLBARS){
        xCorSMA1 = sma(nLength, xSource1);
        xCorSMA2 = sma(nLength, xSource2);
    }

    var nSX = xCorSMA1.getValue(0);
    var nSY = xCorSMA2.getValue(0);

    if (nSX == null || nSY == null)
        return;

    var nCor1 = null;
    var nCor2 = null;
    var nCor3 = null;

    for (var i = 0; i < nLength; i++){
        var nX = xSource1.getValue(-i);
        var nY = xSource2.getValue(-i);
        nCor1 += (nX - nSX) * (nY - nSY);
        nCor2 += (nX - nSX) * (nX - nSX);
        nCor3 += (nY - nSY) * (nY - nSY);
    }

    var nCorrelationRes = nCor1 / Math.sqrt(nCor2) / Math.sqrt(nCor3);

    return nCorrelationRes;
}

function verify(){
    var b = false;
    if (getBuildNumber() < 3742){
```

```
        drawTextAbsolute(5, 35, "This study requires version 12.1 or later.",
            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,
            null, 13, "error");
        drawTextAbsolute(5, 20, "Click HERE to upgrade.@URL=http://www.esignal.com
/download/default.asp",
            Color.white, Color.blue,
Text.RELATIVETOBOTTOM|Text.RELATIVETOLEFT|Text.BOLD|Text.LEFT,
            null, 13, "upgrade");
        return b;
    }
    else
        b = true;

    return b;
}
```

—Eric Lippert
eSignal, an Interactive Data company
800 779-6555, www.eSignal.com

BACK TO LIST

THINKORSWIM: DECEMBER 2015

In “Trading The Loonie” in this issue, author Markos Katsanos explains the heavy correlation between the Canadian dollar and crude oil. He then goes on to describe how one could trade this correlation. Using similar logic as that employed in Bollinger Bands, Katsanos has built a study to provide buy and sell signals for trading the Canadian dollar future.

We have replicated his BBDivergence study and strategy using our proprietary scripting language, *thinkscript*. We have made the loading process extremely easy: simply click on the links http://tos.mx/v0vYZd and http://tos.mx/rdRUvt and choose “Save script to thinkorswim,” then “Backtest in thinkScript.” Choose to rename your study and strategy as “BBDivergence.” You can adjust the parameters of this study within the *edit studies* window to fine-tune your variables.

In the example shown in Figure 2, you see a chart of /6C, which is the Canadian dollar future. Just below the volume and open interest indicator is Katsanos’s BBDivergence indicator, which is used to define the conditions that trigger the trade plotted. Finally, in the bottom indicator, you can see that over this four-year charted time frame, the BBDivergence strategy has produced a positive return.

FIGURE 2: THINKORSWIM. This example chart shows the Canadian dollar future.

For more on this indicator, please see Katsanos's article.

—thinkorswim
A division of TD Ameritrade, Inc.
www.thinkorswim.com

BACK TO LIST

WEALTH-LAB: DECEMBER 2015

This Traders' Tip is based on "Trading The Loonie" in this issue by Markos Katsanos.

The CAD/USD, also known as the "loonie," is a petrocurrency like NOK/USD, USD/RUB and others. The price of oil undoubtedly plays the major role in its exchange rate for the most part. However, the correlation with oil is not always a positive 1.0, since other drivers like the key rate or carry trading periodically come into play.

The system is most likely to exhibit periods of prosperous behavior while crude oil is volatile and trending, and is expected to take a break during prolonged range-bound movements like in 2011–2013.

In addition to the author's use of the light crude oil (CLC) contract, it may also make sense to experiment with the WTI contract to measure correlation. Some believe that this blend follows Canada's Western Canadian Select more accurately. We leave testing this idea to motivated traders.

A participating indicator, *correlation*, is part of our Community Indicators library, w hich is driven by the Wealth-Lab user community. To run the Wealth-Lab 6 strategy code show n here, install the indicator library (or update to the actual version using the Extension Manager) from our w ebsite, w ealth-lab.com, in the Extensions section.

After updating the library to v2015.07 or later, the correlation indicator can be found under the Community Indicators group. Applying it to charts or rule-based strategies is as easy as drag & drop w ithout having to program any code yourself.

A sample chart is show n in Figure 3.

FIGURE 3: WEALTH-LAB. Signals generated by the intermarket Bollinger Bands (BB) divergence method are superimposed on the chart as colored triangles. The 20-day BB divergence is plotted on the top and crude oil (light sweet) futures in the middle window.

Wealth-Lab 6 strategy code (C#):

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Drawing;
using WealthLab;
using WealthLab.Indicators;
using Community.Indicators;

namespace WealthLab.Strategies
{
        public class LoonieStrategy : WealthScript
        {
                private StrategyParameter paramPeriod;

                public LoonieStrategy()
```

```
                {
                        paramPeriod = CreateParameter("days", 20, 2, 300, 1);
                }

                protected override void Execute()
                {
                        int days = paramPeriod.ValueInt;
                        StdDevCalculation s = StdDevCalculation.Sample;
                        var oil = GetExternalSymbol("Light",true).Close;
                        var bbCAD = 1 + ((Close - SMA.Series(Close,days) +
2*StdDev.Series(Close,days,s)) / (4*StdDev.Series(Close,days,s)+0.0001));
                        var bbOil = 1 + ((oil - SMA.Series(oil,days) +
2*StdDev.Series(oil,days,s)) / (4*StdDev.Series(oil,days,s)+0.0001));
                        var divergence =(bbOil-bbCAD)/bbCAD*100;
divergence.Description = "Divergence";

                        var hhvDiv = Highest.Series(divergence,3);
                        var llvDiv = Lowest.Series(divergence,3);
                        var llv = Lowest.Series(Low,15);
                        var hhv = Highest.Series(High,15);
                        var roc = ROC.Series(Close,2);
                        var mov = SMA.Series(oil,40);
                        var correlation20 = Correlation.Series(Close,oil,20);
                        var correlation60 = Correlation.Series(Close,oil,60);
                        var rocOil = ROC.Series(oil,3);
                        var llvOil = Lowest.Series(oil,4);
                        var macd = MACD.Series(Close);
                        var macdSignal = EMAModern.Series(macd,9);
                        var sto = StochD.Series(Bars,30,3);

                        ChartPane oilPane = CreatePane(50,true,true); HideVolume();

PlotSymbol(oilPane,GetExternalSymbol("Light",true),Color.Green,Color.Red);
                        ChartPane divPane = CreatePane(50,true,true);
                        LineStyle ld = LineStyle.Dashed;
                        PlotSeries(divPane,divergence,Color.Black,LineStyle.Solid,1);
                        DrawHorzLine(divPane,20,Color.Black,ld,1);
                        DrawHorzLine(divPane,-20,Color.Black,ld,1);

                        for(int bar = GetTradingLoopStartBar(60); bar < Bars.Count;
bar++)
                        {
                                if (IsLastPositionActive)
                                {
                                        bool sell =
                                                (CrossUnder(bar, macd, macdSignal) &&
sto[bar]>85) ||
                                                (llvDiv[bar] < -20 && rocOil[bar] <
-3) ||
                                                (Close[bar] < llv[bar-1] &&
correlation60[bar] < -0.4);

                                        bool cover =
                                                (CrossOver(bar, macd, macdSignal) &&
sto[bar]<25 &&
                                                (oil[bar] >= (1 + 4.0/100d) *
llvOil[bar]) ) ||
                                                (hhvDiv[bar]>20 && rocOil[bar]>4.5 )
||
                                                (Close[bar] > hhv[bar-1] &&
```

```
correlation60[bar] < -0.4);

                                        Position p = LastPosition;
                                        if( p.PositionType == PositionType.Long )
                                        {
                                                if( sell )
                                                        SellAtMarket(bar+1, p);
                                        }
                                        else
                                        {
                                                if( cover )
                                                        CoverAtMarket(bar+1, p);
                                        }
                                }
                                else
                                {
                                        bool buy = (hhvDiv[bar] > 20) &&
(divergence[bar] < divergence[bar-1]) &&
                                                (roc[bar]>0) && (mov[bar] >
mov[bar-2]) && (correlation20[bar] > -0.4);

                                        bool shrt = (llvDiv[bar] < -20) &&
(divergence[bar] > divergence[bar-1]) &&
                                                (roc[bar]<0) && (mov[bar] <
mov[bar-2]) && (correlation20[bar] > -0.4);

                                        if( buy )
                                                BuyAtMarket(bar+1);
                                        else
                                                if( shrt )
                                                        ShortAtMarket(bar+1);
                                }
                        }
                }
        }
}
```

—Eugene, Wealth-Lab team
MS123, LLC
www.wealth-lab.com

BACK TO LIST

AMIBROKER: DECEMBER 2015

In “Trading The Loonie” in this issue, author Markos Katsanos presents a trading system for CAD futures based on its relationship w ith price action on crude oil futures.

A ready-to-use formula for use in AmiBroker is show n here. To use the Bollinger Band divergence oscillator, enter the code in the formula editor and press the *apply indicator* button. You can adjust the averaging period using the parameters w indow . To backtest the system, click the *send to analysis* button in the formula editor and then the *backtest* button in the *analysis* w indow . You may need to change the symbol of crude oil futures to match your data provided symbology (the code show n here uses IQFeed symbology).

A sample chart is show n in Figure 4.

FIGURE 4: AMIBROKER. Here is a daily chart of CAD futures (upper pane) with buy (green), sell (red), short (hollow red), and cover (hollow green) arrows generated by the system. The Bollinger Band divergence oscillator is shown in the bottom pane.

AmiBroker code:

```
// using IQFeed symbology here
// you may need to adjust symbol for different data source
SEC2 = Foreign( "QCL#", "C" ); // crude oil futures continuous
D1 = Param( "BB DAYS " , 20, 1, 200, 1 );

sec1BOL = 1 + ( ( C - MA( C, D1 ) + 2 * StDev( C, D1 ) ) /
                 ( 4 * StDev( C, D1 ) + 0.0001 ) );
sec2BOL = 1 + ( ( SEC2 - MA( SEC2, D1 ) + 2 * StDev( SEC2, D1 ) ) /
                 ( 4 * StDev( SEC2, D1 ) + 0.0001 ) );
DIV1 = ( sec2BOL - sec1BOL ) / sec1bol * 100;

Plot( Div1, "Bollinger Band Divergence", colorRed );

Buy =  HHV( DIV1, 3 ) > 20 AND
```

```
        DIV1 < Ref( DIV1, -1 ) AND
        ROC( C, 2 ) > 0 AND MA( SEC2, 40 ) > Ref( MA( SEC2, 40 ), -2 )
        and
        Correlation( C, SEC2, 20 ) > -0.4;

Filter = 1;
AddColumn( Buy, "Buy" );

Sell = ( Cross( Signal(), MACD() )
         AND StochK( 30, 3 ) > 85 ) OR
       ( LLV( DIV1, 3 ) < -20 AND ROC( SEC2, 3 ) < -3 ) OR
       ( C < Ref( LLV( L, 15 ), -1 ) AND
         Correlation( C, SEC2, 60 ) < -0.4 );


Short =  LLV( DIV1, 3 ) < -20 AND
         DIV1 > Ref( DIV1, -1 ) AND
         ROC( C, 2 ) < 0 AND
         MA( SEC2, 40 ) < Ref( MA( SEC2, 40 ), -2 ) AND
         Correlation( C, SEC2, 20 ) > -0.4;


Cover =  ( Cross( MACD(), Signal() ) AND StochK( 30, 3 ) < 25 AND
           SEC2 >= ( 1 + 4 / 100 ) * LLV( SEC2, 4 ) ) OR
         ( HHV( DIV1, 3 ) > 20 AND ROC( SEC2, 3 ) > 4.5 ) OR
         ( C > Ref( HHV( High, 15 ), -1 ) AND
             Correlation( C, SEC2, 60 ) < -0.4 );
```

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

BACK TO LIST

NEUROSHELL TRADER: DECEMBER 2015

The Bollinger Band divergence indicator described by Markos Katsanos in his article in this issue, “Trading The Loonie,” can be easily implemented w ith a few of NeuroShell Trader’s 800+ indicators. Simply select *new indicator* from the *insert* menu and use the indicator w izard to set up the follow ing indicator:

```
BBDIV
Multiply2( 100, Divide( Sub( BB%B(Crude Oil Close,20,2), BB%B(Close,20,2)), Add2(100,
BB%B( Close,20,2))))
```

To implement the Canadian loonie and crude oil divergence trading system, simply select *new trading strategy* from the *insert* menu and enter the follow ing formulas in the appropriate locations of the trading strategy w izard:

BUY LONG CONDITIONS: [All of w hich must be true]

```
A>B( Max(BBDIV(Close,Crude Oil Close,20,2),3),20)
A<B( Momentum(BBDIV(Close,Crude Oil Close,20,2),1),0)
A>B( %Change(Close,2),0)
A>B( Momentum(Avg(Crude Oil Close,40),2),0)
A>B( LinXYReg r(Close,Crude Oil Close,20),-0.4)
```

SELL LONG CONDITIONS: [1 of w hich must be true]

```
And2( CrossAbove(MACD Signal(Close,9,12,26), MACD(Close,12,26)), A>B( Stoch%D( High,
Low, Close, 30, 3), 85))
And2( A<B(Min(BBDIV(Close,Crude Oil Close,20,2),3),-20), A<B( %Change(Close,3),-3))
And2( A<B(Close,Lag(PriceLow(Low,15),1)),A<B(LinXYReg r(Close,Crude Oil Close,20),
-0.4))
```

SELL SHORT CONDITIONS: [All of w hich must be true]

```
A<B( Min(BBDIV(Close,Crude Oil Close,20,2),3),-20)
A>B( Momentum(BBDIV(Close,Crude Oil Close,20,2),1),0)
A<B( %Change(Close,2),0)
A<B( Momentum(Avg(Crude Oil Close,40),2),0)
A>B( LinXYReg r(Close,Crude Oil Close,20),-0.4)
```

COVER SHORT CONDITIONS: [1 of w hich must be true]

```
And3( CrossBelow(MACD Signal(Close,9,12,26), MACD(Close,12,26)), A<B( Stoch%D( High,
Low, Close, 30, 3), 25), A>=B(Crude Oil Close, Add2(1, Mul2(0.04, Min(Crude Oil
Close,4)))))
And2( A>B(Max(BBDIV(Close,Crude Oil Close,20,2),3),20), A>B( %Change(Close,3),4.5))
And2( A>B(Close,Lag(PriceHigh(High,15),1)),A<B(LinXYReg r(Close,Crude Oil
Close,20),-0.4))
```

Users of NeuroShell Trader can go to the Stocks & Commodities section of the NeuroShell Trader free technical support w ebsite to dow nload a copy of this or any previous Traders' Tips.

A sample chart is show n in Figure 5.

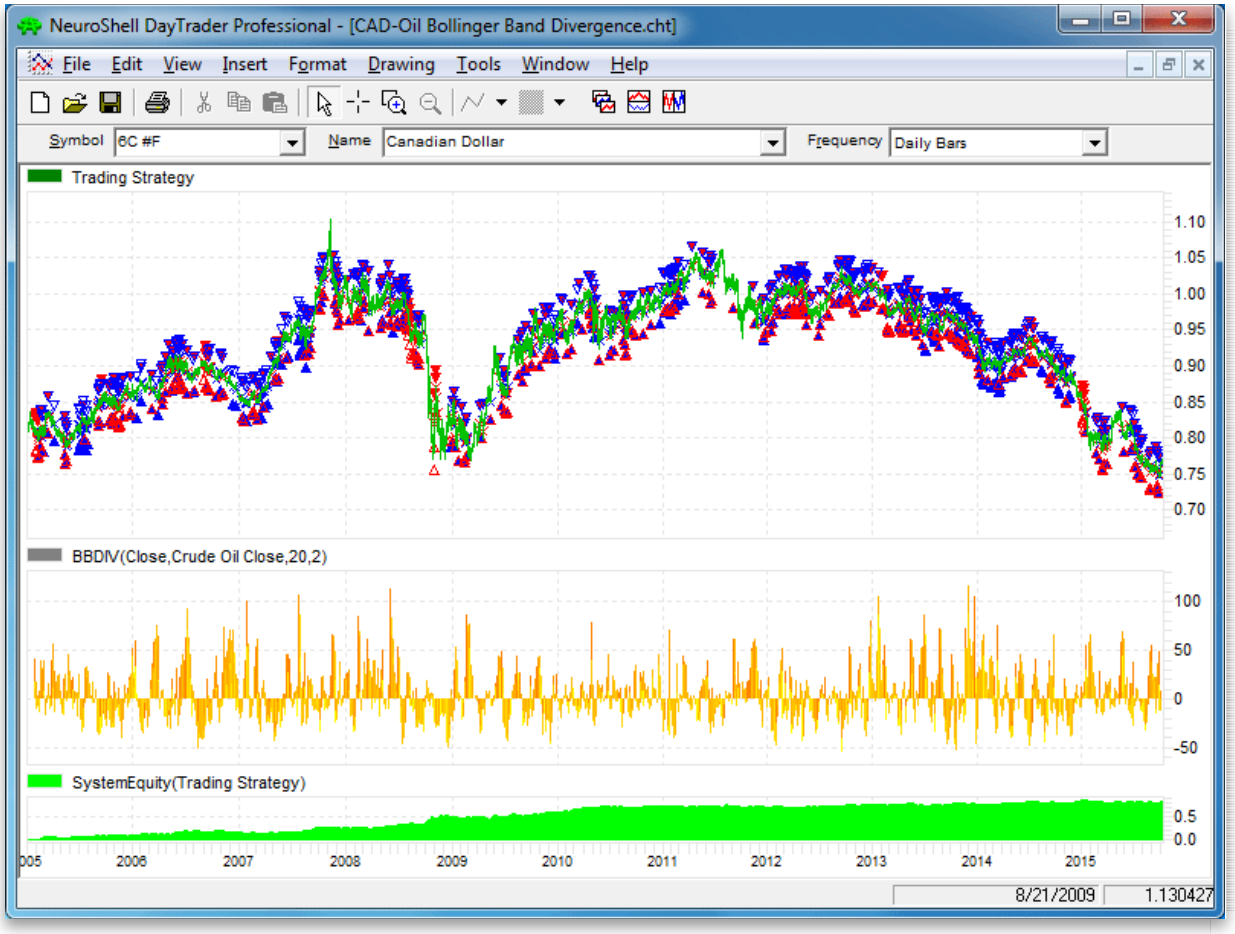


FIGURE 5: NEUROSHELL TRADER. This NeuroShell Trader chart shows the Canadian loonie and crude oil Bollinger Band divergence system.

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

BACK TO LIST

AIQ: DECEMBER 2015

Here is some code for use in AIQ based on Markos Katsanos's article in this issue, "Trading The Loonie." The code and EDS file can be downloaded from www.TradersEdgeSystems.com/traderstips.htm.

The code I am providing contains both the divergence indicator and a long-only trading system for the NASDAQ 100 list of stocks. Rather than trading forex, I wanted to try the divergence idea and the author's entry rules on the NASDAQ 100 stocks. The stocks are traded long using the author's entry rules with two of the parameters adjusted as shown at the top of the code file. The exit has been changed completely to use a

profit protect (protect 50% of profits once a 20% profit is reached), a stop-loss (protect 75% of capital), and a time-stop exit (exit after 21 days). I used the NASDAQ 100 index (NDX) in place of the crude oil futures. The assumption is that since the stocks on the list are all in the NDX, they w ould generally be correlated to the index. The author's entry rule filters out those w ith a negative correlation to the index. Note that I changed the minimum correlation from a -0.4 to 0.0. In addition, I found that increasing the minimum divergence from 20 to 2,000 increased the Sharpe ratio and decreased the maximum draw dow n w ithout affecting the annualized return.

Figure 6 show s the equity curve versus the NASDAQ 100 index for the period 1/5/2000 to 10/14/2015. Figure 7 show s the metrics for this same test period. The system clearly outperformed the index.

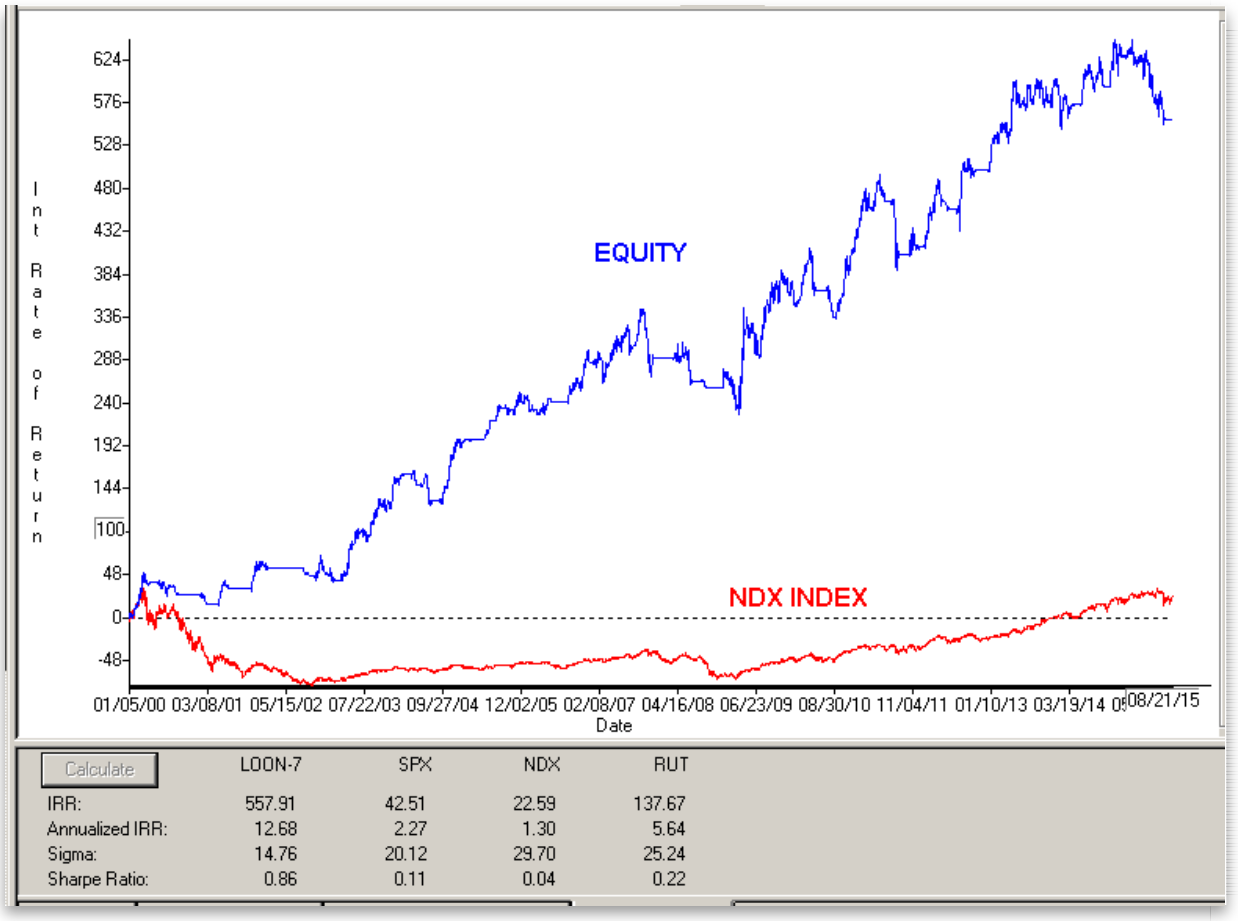


FIGURE 6: AIQ. Here is a sample equity curve for the modified divergence system versus the NASDAQ 100 index for the period 1/5/2000 to 10/14/2015.

Account Statistics/Analysis

Start date:	01/05/00			
End date:	10/14/15			
		Winners	Losers	Neutral
		==========	==========	==========
Number of trades:	1044	586	458	0
Average periods per trade:	29.90	29.83	29.99	0.00
Maximum Profit/Loss:		149.11 %	(35.40)%	
Average Drawdown:	(5.55)%	(2.46)%	(9.51)%	
Average Profit/Loss:	2.04 %	9.69 %	(7.74)%	
Average NDX Profit/Loss:	0.17 %	1.76 %	(1.85)%	
Probability:		56.13 %	43.87 %	
Average Annual ROI:	24.94 %	118.52 %	(94.17)%	
Annual NDX (Buy & Hold):	1.46 %			
Reward/Risk Ratio:	1.60			

Portfolio:

Starting Balance:	500000.00		
Ending Balance:	3289564.83		
Gain/Loss:	2789564.83		
Gain/Loss (IRR) %:	557.91 %	Annualized:	12.68 %

Drawdown:		From	To
Maximum Continuous:	(10.04)%	08/03/11	08/08/11
Portfolio High:	(1.15)%	11/28/14	12/02/14
Peak/Valley:	(26.51)%	10/05/07	03/20/09
Initial Investment:	(0.91)%		

Account default strategy:
Buy 21day exit-stop loss loose-p
Using list: NAS100
Pricing Summary:
Entry price: [Open]
Exit price: [Open]
Entry Summary:
Buy
DIVERG - Descending
DIVERG1 - Descending

Exit Summary:
Hold for 21 periodsams tight
Cap. Prot. 75%
Profit Prot. 50% above 20%

Capitalization
10.00% of portfolio value
Computed every 1 days
Partial entry, 100 minimum shares
No more than 3 new trades per day
No more than 10 open positions
Leverage = 0%
Initial requirement for short positions = 50%

File C:\wintes32\Eds Strategies_TASC AIQ\2015\12_LOONIE\Working\LOONIE.EDS

FIGURE 7: AIQ. Here are the metrics for the modified system and the test settings.

```
!TRADING THE LOONIE
!Author: Markos Katsanos, TASC December 2015
!coded by: Richard Denning 10/17/15
!www.TradersEdgeSystems.com

!Set parameters:
 Define Len        20. !Default is 20
 Define F1          2. !Default is 2
 Define F2          4. !Default is 4
 IDX is         "NDX". !NASDAQ 100 index
 IDXsLen is        40. !Default is 40
```

```
 minDIVERG is    2000. !Default is 20
 minROC is          0. !Default is 0
 minCorrel is     0.0. !Default is -0.4

!Close percent relative to BB band width for stock:
Variance is Variance([close],Len).
StdDev is Sqrt(Variance).
SMA is simpleavg([close],Len).
stkBB is 1+([close]-SMA+F1*StdDev)/(F2*StdDev).

!Close percent relative to BB band width for index:
IDXc is tickerUDF(IDX,[close]).
VarianceIdx is Variance(IDXc,Len).
StdDevIDX is Sqrt(Variance).
SMAidx is simpleavg(IDXc,Len).
idxBB is 1+(IDXc-SMAidx+F1*StdDevIDX)/(F2*StdDevIDX).

DIVERG is (idxBB-stkBB)/stkBB*100.     !PLOT AS CUSTOM INDICATOR
DIVERG1 is valresult(DIVERG,1).
ROC2 is ([close]/val([close],2)-1)*100.
ROC3 is ([close]/val([close],3)-1)*100.
ROC3idx is tickerUDF(IDX,ROC3).
IDXsma is simpleavg(IDXc,IDXsLen).
IDXsma2 is valresult(IDXsma,2).
HHVdiverg is highresult(DIVERG,3).

Setup1 if highresult(DIVERG,3) > minDIVERG.
Setup2 if DIVERG < valresult(DIVERG,1).
Setup3 if ([close]/val([close],2)-1)*100 > minROC.
Setup4 if IDXsma > valresult(IDXsma,2).
Setup5 if pCorrel > minCorrel.

Buy if  Setup1 and
        Setup2 and
        Setup3 and
        Setup4 and
        Setup5.

BuyAlt if Buy.

LongExit1 if MACD<sigMACD and valrule(MACD>sigMACD,1) and
             Stoch > 85.
LongExit2 if lowresult(DIVERG,3)<-20 and ROC3idx<-0.4.
LongExit3 if [close]<loval([close],15,1) and pCorrel<minCorrel.
LongExit if LongExit1 or LongExit2 or LongExit3.

AlterLongExit if {position days} >=21 or [close] <= (1-0.25)*{position entry price}.

!Code to Calculate Pearson's R [for entry]:
! PeriodtoTest is the number of lookback days.
! IndexTkr is the Instrument that you which to compare your list to.
PeriodToTest is Len.
IndexTkr is IDX.
ChgTkr is ([open] / val([open],PeriodToTest)-1)*100.
ChgIdx is TickerUDF(IndexTkr,ChgTkr).
Alpha is ChgTkr - ChgIdx.

ValUDF is (([close]-[open])/[open]) * 100.
ValIndex is TickerUDF(IndexTkr, ValUDF).
ValTkr is ValUDF.
```

```
SumXSquared is Sum(Power(ValIndex,2), PeriodToTest).
SumX is Sum(ValIndex, PeriodToTest).
SumYSquared is Sum(Power(ValTkr,2), PeriodToTest).
SumY is Sum(ValTkr, PeriodToTest).
SumXY is Sum(ValTkr*ValIndex, PeriodToTest).
SP is SumXY - ( (SumX * SumY) / PeriodToTest ).
SSx is SumXSquared - ( (SumX * SumX) / PeriodToTest ).
SSy is SumYSquared - ( (SumY * SumY) / PeriodToTest ).

!Pearson's R and Pearson's Coefficient of Determination:
pCorrel is SP/SQRT(SSX*SSY).

!Code to Calculate Pearson's R [for exit]:
! PeriodtoTest is the number of lookback days.
! IndexTkr is the Instrument that you which to compare your list to.
PeriodToTestX is 3*Len.
IndexTkrX is IDX.
ChgTkrX is ([open] / val([open],PeriodToTestX)-1)*100.
ChgIdxX is TickerUDF(IndexTkrX,ChgTkrX).
AlphaX is ChgTkrX - ChgIdxX.

ValUDFX is (([close]-[open])/[open]) * 100.
ValIndexX is TickerUDF(IndexTkrX, ValUDFX).
ValTkrX is ValUDFX.
SumXSquaredX is Sum(Power(ValIndexX,2), PeriodToTestX).
SumXX is Sum(ValIndexX, PeriodToTestX).
SumYSquaredX is Sum(Power(ValTkrX,2), PeriodToTestX).
SumYX is Sum(ValTkrX, PeriodToTestX).
SumXYX is Sum(ValTkrX*ValIndexX, PeriodToTestX).
SPX is SumXYX - ( (SumXX * SumYX) / PeriodToTestX).
SSxX is SumXSquaredX - ( (SumXX * SumXX) / PeriodToTestX ).
SSyX is SumYSquaredX - ( (SumYX * SumYX) / PeriodToTestX ).

!Pearson's R and Pearson's Coefficient of Determination:
pCorrelX is SPX/SQRT(SSXX*SSYX).

!MACD code:
S is 12.
L is 25.
X is 9.

ShortMACDMA is expavg([Close],S).
LongMACDMA is expavg([Close],L).

MACD is ShortMACDMA-LongMACDMA.
SigMACD is expavg(MACD,X).

!Stochastic
StochLen is 30.
Stoch is 100 * (([Close]-LoVal([Low],StochLen)) /
        (HiVal([High],StochLen) - LoVal([Low],StochLen))).

List if 1.
```

—Richard Denning
info@TradersEdgeSystems.com
for AIQ Systems

BACK TO LIST

TRADERSSTUDIO: DECEMBER 2015

The TradersStudio code based on Markos Katsanos's article in this issue, "Trading the Loonie," can be found at www.TradersEdgeSystems.com/traderstips.htm.

The following code files are provided in the download:

- Function DIVERG—Computes the divergence indicator for the tradable and a another security, index or future
- Indicator plot DIVERG_IND—For plotting the DIVERG on a chart.

Figure 8 shows the DIVERG indicator on a chart of Apple Inc.

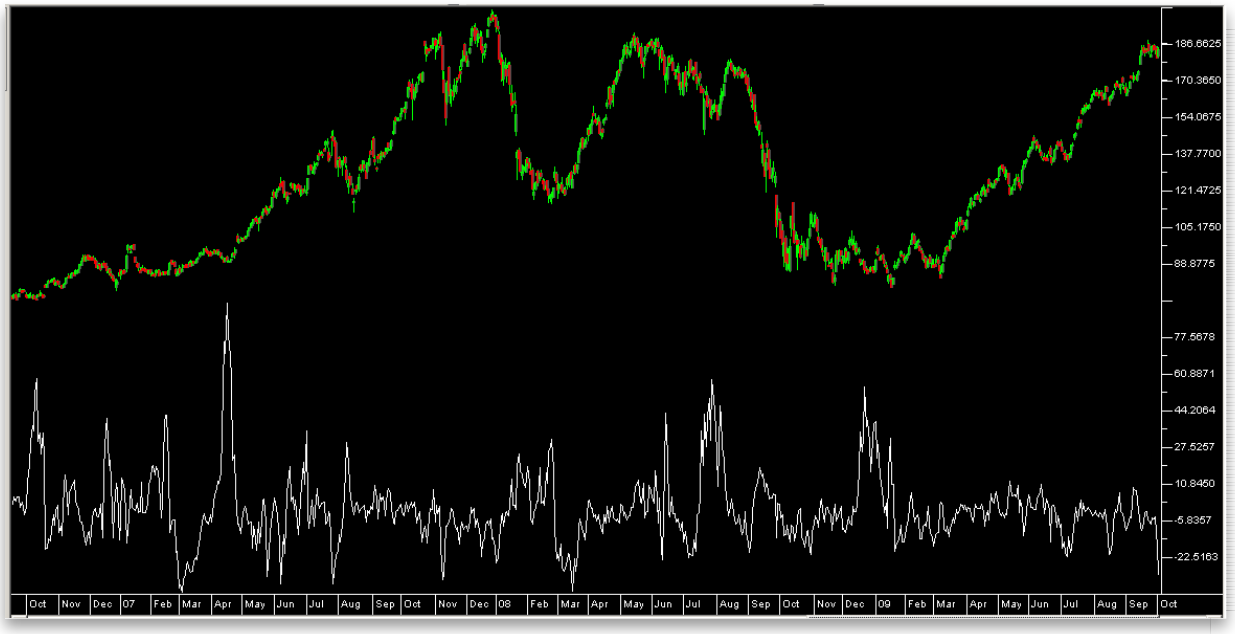


FIGURE 8: TRADERSSTUDIO. Here is an example of the DIVERG indicator on a chart of Apple Inc.

```
'TRADING THE LOONIE
'Author: Markos Katsanos, TASC December 2015
'Coded by: Richard Denning 10/19/15
'www.TradersEdgeSystems.com

function DIVERG(bbLen,F1,F2)
'Set parameters:
 'bbLen = 20. 'Default = 20
 'F1 = 2 Default = 2
 'F2 = 4 Default = 4
 'IDX = "NDX" NASDAQ 100 Index for independent 1

'Close percent relative to BB band width for stock:
Dim SD As BarArray
```

```
Dim SMA As BarArray
Dim stkBB As BarArray
SD = StdDev(C,bbLen)
SMA = Average(C,bbLen)
stkBB = 1+(C-SMA+F1*SD)/(F2*SD)

'Close percent relative to BB band width for index:
Dim IDXc As BarArray
Dim StdDevIDX As BarArray
Dim SMAidx As BarArray
Dim idxBB As BarArray
IDXc = C Of independent1
StdDevIDX = StdDev(idxc,bbLen)
SMAidx = Average(IDXc,bbLen)
idxBB = 1+(IDXc-SMAidx+F1*StdDevIDX)/(F2*StdDevIDX)

DIVERG = (idxBB-stkBB)/stkBB*100
end function
'-----------------------------------------------------
'INDICATOR TO PLOT DIVERG:

sub DIVERG_IND(bbLen,F1,F2)
plot1(DIVERG(bbLen,F1,F2))

End Sub
```

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

NINJATRADER: DECEMBER 2015

The TradingTheLoonie strategy and the Bollinger Band divergence indicator, which are discussed in Markos Katsanos's article in this issue, "Trading The Loonie," have been made available for download at www.ninjatrader.com/SC/December2015SC.zip.

Once you have downloaded them, from within the NinjaTrader Control Center window, select the menu File → Utilities → Import NinjaScript and select the downloaded file. This file is for NinjaTrader Version 7.

You can review the strategy source code by selecting the menu Tools → Edit NinjaScript → Strategy from within the NinjaTrader Control Center window and selecting the "TradingtheLoonie" file. To view the indicator source code, go to Tools → Edit NinjaScript → Indicator and select the "BollingerBandDivergence" file.

NinjaScript uses compiled DLLs that run native, not interpreted, to provide the highest performance possible.

A sample chart implementing the strategy is shown in Figure 9.

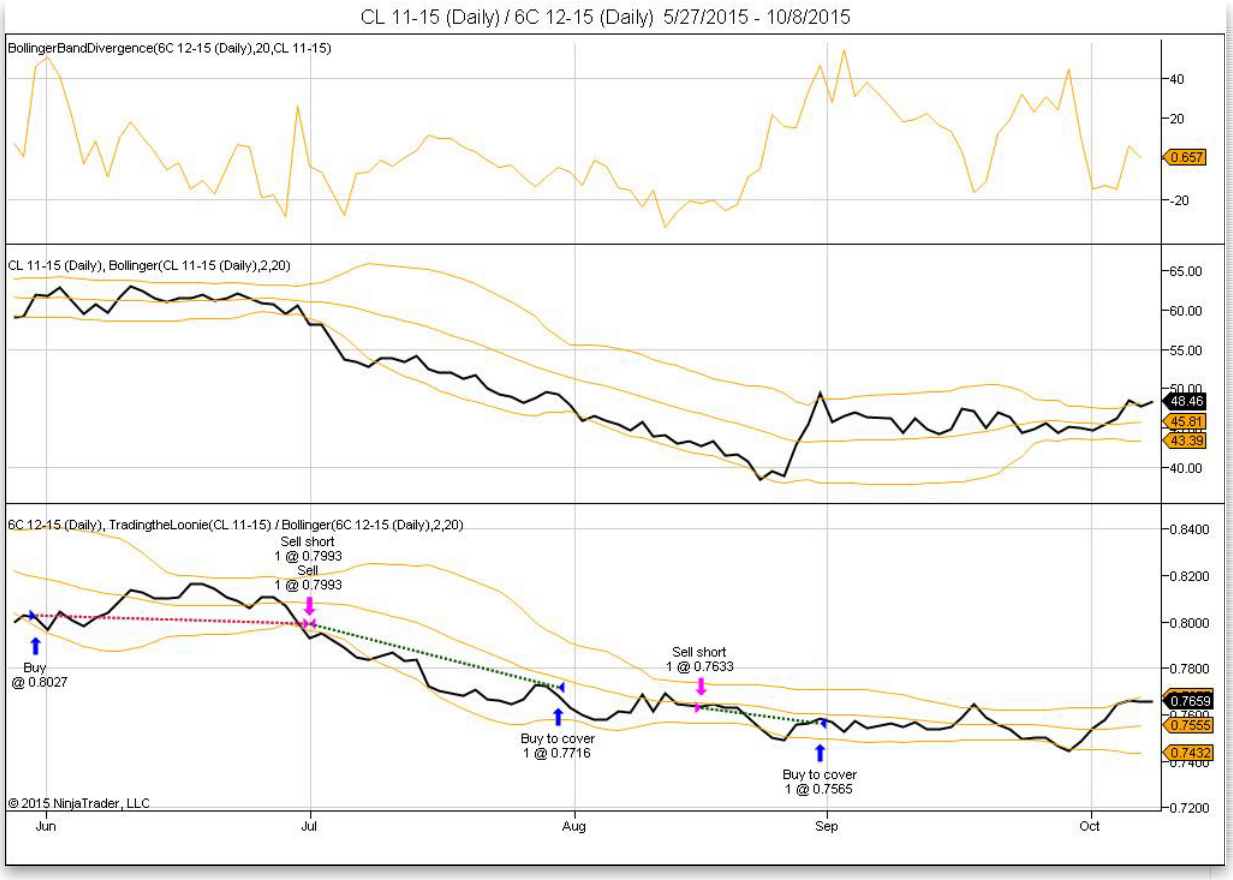


FIGURE 9: NINJATRADER. The Bollinger Band divergence is displayed above the CL and 6C, both of which display the Bollinger Bands in their respective panels. The TradingtheLoonie strategy is displayed in the 6C panel of the chart.

—Raymond Deux and Zachary Gauld
NinjaTrader, LLC
www.ninjatrader.com

BACK TO LIST

UPDATA: DECEMBER 2015

Our Traders' Tip this month is based on "Trading The Loonie" by Markos Katsanos in this issue.

The author proposes that CAD/USD is a commodity currency, meaning it is highly correlated w ith the price of oil. The reason for this is that Canada's main export destination is the US. Thus, w hen the price of oil is high, more US dollars w ill be flow ing into the Canadian economy, increasing the value of the Canadian dollar. Conversely, w hen the price of oil is low , there w ill be a fall in the Canadian dollar value. This trading system primarily uses Bollinger Band divergences to judge the most opportune times to enter into a CAD futures trade.

The Updata code for this article is in the Updata library and may be dow nloaded by clicking the *custom* menu and *system library*. Those w ho cannot access the library due to a firew all can paste the code show n below into the Updata custom editor and save it.

```
'CAD Trading System
Parameter "Bollinger Period" #PERIOD=20
Parameter "Bollinger Dev." @DEV=2
PARAMETER "Ticker" ~TICKER=SELECT
PARAMETER "Donchian Entry Period" #HHLLPeriodEnt=3
PARAMETER "Donchian Exit Period" #HHLLPeriodExt=4
PARAMETER "ROC Period" #ROCPeriod=2
PARAMETER "Average Period" #AVEPeriod=40
PARAMETER "Correl. Entry Period" #CORRPeriodEnt=20
PARAMETER "Correl. Exit Period" #CORRPeriodExt=60
PARAMETER "MACD Ave 1" #MACDPeriod1=12
PARAMETER "MACD Ave 2" #MACDPeriod2=26
PARAMETER "MACD Signal" #MACDSignal=9
PARAMETER "Stochastic 1" #StochasticPeriod1=30
PARAMETER "Stochastic 2" #StochasticPeriod2=3
PARAMETER "|Correl. Thresh.|" @CORRTHRESH=0.4
NAME "BB Divirgence Indicator [" @DEV "|" #PERIOD "]" ""
INDICATORTYPE CHART
DISPLAYSTYLE LINE
@sec1BOLL=0
@sec2BOLL=0
@Div=0
@MACD=0
@MACDSig=0
@STOCHD=0
@CorrelEntry=0
@CorrelExit=0
@UpperDonchEnt=0
@UpperDonchExt=0
@LowerDonchEnt=0
@LowerDonchExt=0
@AvgRef=0
FOR #CURDATE=#PERIOD to #LASTDATE
   'BB Divirgence Indicator
   @sec1BOLL=1+((CLOSE-
SGNL(CLOSE,#PERIOD,M))+2*STDDEV(CLOSE,#PERIOD))/(4*STDDEV(CLOSE,#PERIOD)+0.00001)
   @sec2BOLL=1+((~TICKER-
SGNL(~TICKER,#PERIOD,M))+2*STDDEV(~TICKER,#PERIOD))/(4*STDDEV(~TICKER,#PERIOD)+0.00001
)
   @Div=100*(@sec2BOLL-@sec1BOLL)/@sec1BOLL
   @MACD=MACD(#MACDPeriod1,#MACDPeriod2,E)
   @MACDSig=SGNL(@MACD,#MACDSignal,E)
   @STOCHD=STOCHD(#StochasticPeriod1,#StochasticPeriod2,E)
   @CorrelEntry=BETA(CLOSE,~TICKER,#CORRPeriodEnt)
   @CorrelExit=BETA(CLOSE,~TICKER,#CORRPeriodExt)
   @UpperDonchEnt=PHIGH(@Div,#HHLLPeriodEnt)
   @UpperDonchExt=PHIGH(~TICKER,#HHLLPeriodExt,1)
   @LowerDonchEnt=PLOW(@Div,#HHLLPeriodEnt)
   @LowerDonchExt=PLOW(~TICKER,#HHLLPeriodExt,M)
   @AvgRef=SGNL(~TICKER,#AVEPeriod,M)
   'Long Entries
   IF @UpperDonchEnt>20 AND @Div<HIST(@Div,1) AND MOM(2)>0 AND
@AvgRef>HIST(@AvgRef,1) AND @CorrelEntry>-@CORRTHRESH
```

```
      BUY CLOSE
   ENDIF
   IF HASX(@MACD,@MACDSig,DOWN) AND @STOCHD<25 AND ~TICKER>1.04*PLOW(~TICKER,4)
      COVER CLOSE
   ENDIF
   IF @UpperDonchEnt>20 AND STUDY(~TICKER,MOM(3))>4.5
      COVER CLOSE
   ENDIF
   IF CLOSE>PHIGH(HIGH,15,1) AND @CorrelExit<-@CORRTHRESH
      COVER CLOSE
   ENDIF
   IF @Div>HIST(@Div,1) AND @AvgRef<HIST(@AvgRef,1) AND MOM(2)<0 AND
@LowerDonchEnt<-20 AND @CorrelEntry>-@CORRTHRESH
      SHORT CLOSE
   ENDIF
   IF HASX(@MACD,@MACDSig,UP) AND @STOCHD>85
      SELL CLOSE
   ENDIF
   IF @LowerDonchEnt<-20 AND STUDY(~TICKER,MOM(3))<-3
      SELL CLOSE
   ENDIF
   IF CLOSE<PLOW(LOW,15,1) AND @CorrelExit<-@CORRTHRESH
      SELL CLOSE
   ENDIF
   @PLOT=@Div
NEXT
```

Figure 10 show s an example of the Bollinger Band divergence indicator applied to CAD/USD and crude oil.

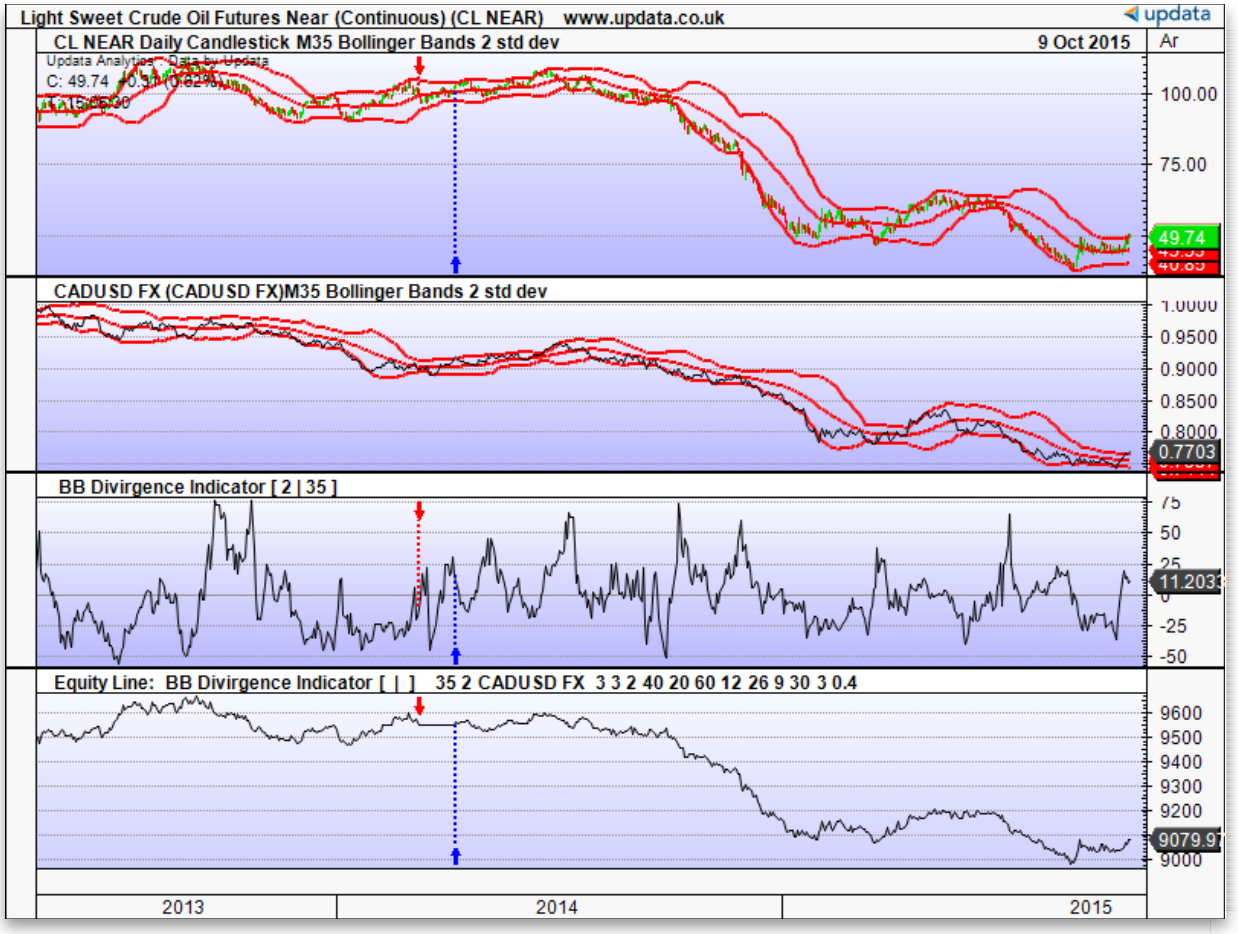


FIGURE 10: UPDATA. Here, the Bollinger Band divergence indicator is applied to CAD/USD and crude oil in daily resolution.

—Updata support team
support@updata.co.uk
www.updata.co.uk

BACK TO LIST

TRADE NAVIGATOR: DECEMBER 2015

We have created a special file based on the formulas given in Markos Katsanos's article in this issue, "Trading The Loonie," to make it easy to dow nload as a library in Trade Navigator. The filename is "SC201512."

To dow nload it, click on Trade Navigator's blue telephone button, select *download special file*, then erase the w ord "upgrade" and type in "SC201512" (w ithout the quotes), then click the *start* button. When prompted to upgrade, click the *yes* button. If prompted to close all softw are, click on the *continue* button. Your library w ill now dow nload.

This library contains four indicators named “CAD Futures TS Buy,” “CAD Futures TS Sell,” “CAD Futures TS Sell Short,” and “CAD Futures TS Cover.” These indicators can be inserted into your chart (Figure 11) by opening the *charting* dropdown menu, then selecting the *add to chart* command, then selecting the *highlight bars* tab.

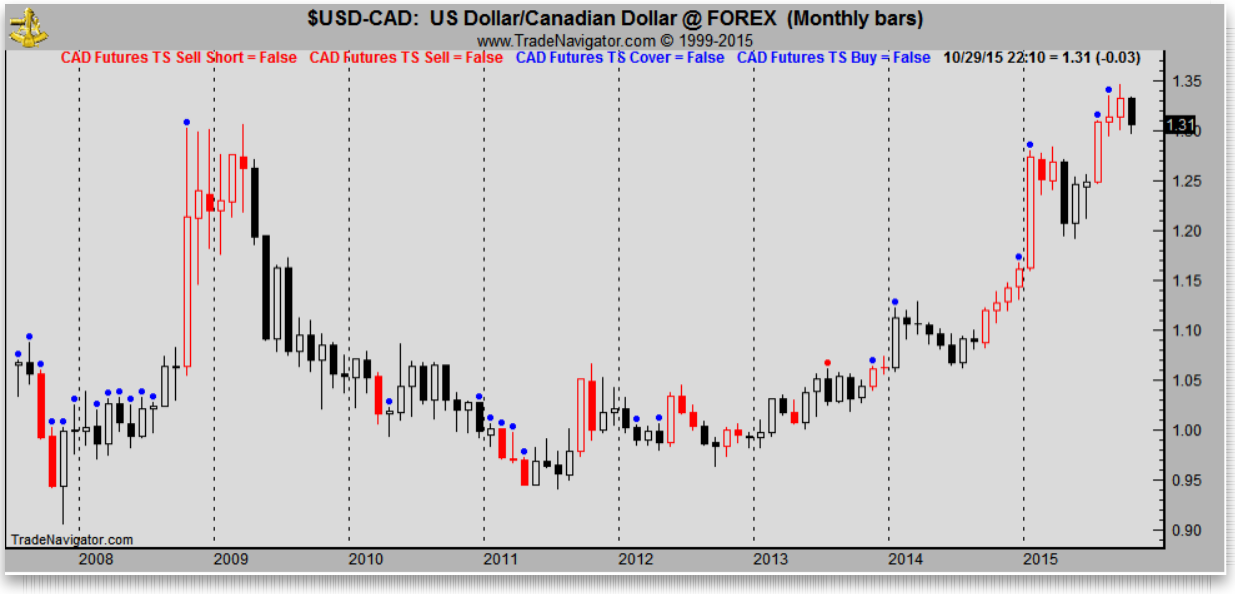


FIGURE 11: TRADE NAVIGATOR. Here, the four indicators (buy, sell, sell short, cover) are applied to a monthly USD/CAD chart as highlight bars.

The TradeSense language for the indicators is as follows:

CAD Futures TS Buy

```
&D1 := 20
&SEC2 := Close Of "CL3-067"
&sec1BOL := 1 + ((Close - MovingAvg (Close , &D1) + 2 * MovingStdDev (Close , &D1)) /
(4 * MovingStdDev (Close , &D1) + .0001))
&sec2BOL := 1 + ((&SEC2 - MovingAvg (&SEC2 , &D1) + 2 * MovingStdDev (&SEC2 , &D1)) /
(4 * MovingStdDev (&SEC2 , &D1) + .0001))
&DIV1 := (&sec2BOL - &sec1BOL) / &sec1bol * 100
Highest (&DIV1 , 3) > 20 And &DIV1 < (&DIV1).1 And RateOfChange (Close , 2) > 0 And
MovingAvg (&SEC2 , 40) > MovingAvg (&SEC2 , 40).2 And Correlation (Close , &SEC2 ,
20) >= (-.4)
```

CAD Futures TS Sell

```
&D1 := 20
&SEC2 := Close Of "CL3-067"
&sec1BOL := 1 + ((Close - MovingAvg (Close , &D1) + 2 * MovingStdDev (Close , &D1)) /
(4 * MovingStdDev (Close , &D1) + .0001))
&sec2BOL := 1 + ((&SEC2 - MovingAvg (&SEC2 , &D1) + 2 * MovingStdDev (&SEC2 , &D1)) /
(4 * MovingStdDev (&SEC2 , &D1) + .0001))
&DIV1 := (&sec2BOL - &sec1BOL) / &sec1bol * 100
(Crosses Above (MovingAvgX (MACD (Close , 12 , 26 , False) , 9 , False) , MACD (Close
, 12 , 26 , False)) And StochK (30 , 3) > 85) Or (Lowest (&DIV1 , 3) < (-20) And
RateOfChange (&SEC2 , 3) < (-3)) Or (Close < Lowest (Low , 15).1 And Correlation
(Close , &SEC2 , 60) < (-.4))
```

CAD Futures TS Sell Short

```
&D1 := 20
&SEC2 := Close Of "CL3-067"
&sec1BOL := 1 + ((Close - MovingAvg (Close , &D1) + 2 * MovingStdDev (Close , &D1)) /
(4 * MovingStdDev (Close , &D1) + .0001))
&sec2BOL := 1 + ((&SEC2 - MovingAvg (&SEC2 , &D1) + 2 * MovingStdDev (&SEC2 , &D1)) /
(4 * MovingStdDev (&SEC2 , &D1) + .0001))
&DIV1 := (&sec2BOL - &sec1BOL) / &sec1bol * 100
Lowest (&DIV1 , 3) < (-20) And &DIV1 > (&DIV1).1 And RateOfChange (Close , 2) < 0 And
MovingAvg (&SEC2 , 40) < MovingAvg (&SEC2 , 40).2 And Correlation (Close , &SEC2 ,
20) > (-.4)
```

CAD Futures TS Cover

```
&D1 := 20
&SEC2 := Close Of "CL3-067"
&sec1BOL := 1 + ((Close - MovingAvg (Close , &D1) + 2 * MovingStdDev (Close , &D1)) /
(4 * MovingStdDev (Close , &D1) + .0001))
&sec2BOL := 1 + ((&SEC2 - MovingAvg (&SEC2 , &D1) + 2 * MovingStdDev (&SEC2 , &D1)) /
(4 * MovingStdDev (&SEC2 , &D1) + .0001))
&DIV1 := (&sec2BOL - &sec1BOL) / &sec1bol * 100
(Crosses Below (MACD (Close , 12 , 26 , False) , MovingAvgX (MACD (Close , 12 , 26 ,
False) , 9)) And StochK (30 , 3) < 25 And &SEC2 >= (1 + 4 / 100) * Lowest (&SEC2 ,
4)) Or (Highest (&DIV1 , 3) > 20 And RateOfChange (&SEC2 , 3) > 4.5) Or (Close >
Highest (High , 15).1 And Correlation (Close , &SEC2 , 60) < (-.4))
```

To create this indicator manually, click on the *edit* dropdow n menu and open the *trader's toolbox* (or use CTRL+T) and click on the *functions* tab. Now click on the *new* button, and a *new function* dialog w indow w ill open. In its text box, type in the code for the highlight bar. Ensure that there are no extra spaces at the end of each line. When completed, click on the *verify* button. You may be presented w ith an *add inputs* popup message if there are variables in the code. If so, click the *yes* button, then enter a value in the *default value* column. If all is w ell, w hen you click on the *function* tab, the code you entered w ill convert to italic font. Click on the *save* button, and type a name for the indicator.

Strategy

This library also contains a strategy called "SC CAD Futures Trading System." This prebuilt strategy can be overlaid on your chart (Figure 12) by opening the charting dropdow n menu, selecting the *add to chart* command, then selecting the *strategies* tab.

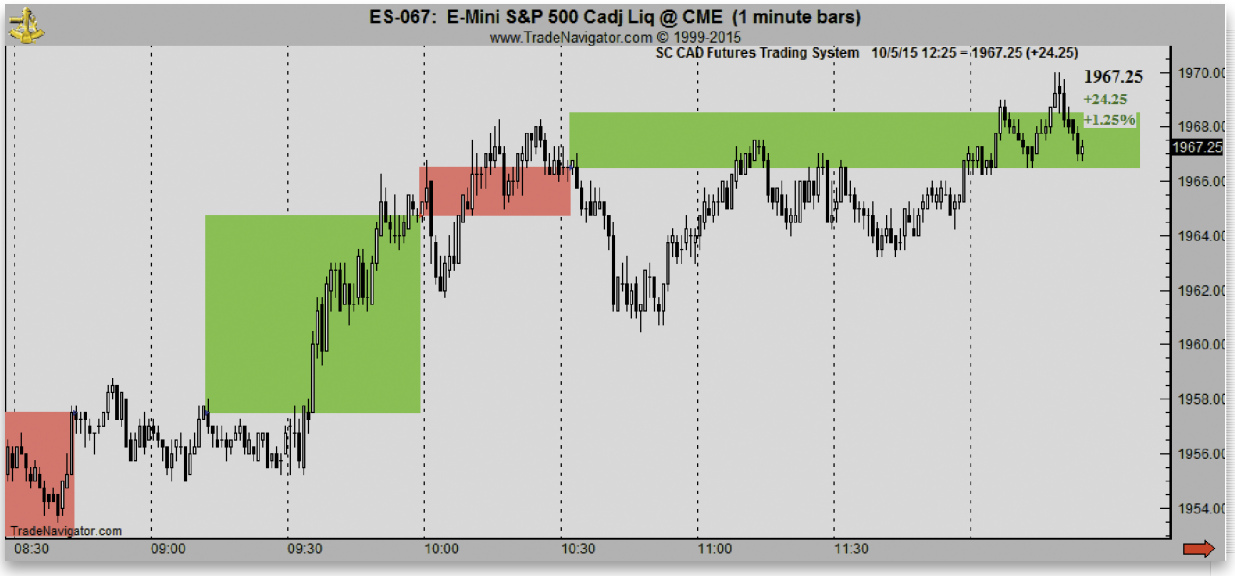


FIGURE 12: TRADE NAVIGATOR. Strategy entry/exit points are displayed on a one-minute USD/CAD chart with profit/loss shading.

If you have any difficulty using the indicators or strategy, you can contact our technical support staff at 719 884-0245 or click on the live chat tool either under Trade Navigator's help menu or near the top of our homepage at www.TradeNavigator.com. Our support hours are 6 am–7 pm (M-F) Mountain Time. Happy trading!

—Genesis Financial Technologies
www.TradeNavigator.com

BACK TO LIST

Originally published in the December 2015 issue of
Technical Analysis of STOCKS & COMMODITIES magazine.