TRADERS' TIPS

January 2014

For this month's Traders' Tips, the focus is mainly John Ehlers' article in this issue, "Predictive And Successful Indicators." Here we present the January 2014 Traders' Tips code with possible implementations in various software.

Code for TradeStation is already provided in Ehlers' article. Subscribers will find that code at the Subscriber Area of our website, www.Traders.com. (Click on "Article Code" from the S&C menu.) Presented here is an overview of possible implementations for other software.

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

TRADESTATION: JANUARY 2014 TRADERS' TIPS CODE

In "Predictive And Successful Indicators" in this issue, author John Ehlers describes a new method for smoothing market data while reducing the lag that most other smoothing techniques have. Ehlers has provided some TradeStation code in his article for several indicators and describes an approach for creating a strategy. For convenience, we will make the same code available at our EasyLanguage support forum as well as an example strategy based on Ehlers' description.

To download the EasyLanguage code, please visit our TradeStation and EasyLanguage support forum. The code can be found here: www.tradestation.com/TASC-2014. The ELD filename is "_TASC_PredictiveIndicators.ELD." For more information about EasyLanguage in general, please see www.tradestation.com/EL-FAQ.

The code is also shown below.

```
_Ehlers_SuperSmoother Filter (Indicator)

variables:
        a1( 0 ),
        b1( 0 ),
        c1( 0 ),
        c2( 0 ),
        c3( 0 ),
        Filt( 0 ) ;

a1 = ExpValue( -1.414 * 3.14159 / 10 ) ;
b1 = 2 * a1 * Cosine( 1.414 * 180 / 10 ) ;
c2 = b1 ;
c3 = -a1 * a1 ;
c1 = 1 - c2 - c3 ;
Filt = c1 * ( Close + Close[1] )
        / 2 + c2 * Filt[1] + c3 * Filt[2] ;

Plot1( Filt, "SuperSmooth" ) ;


_Ehlers_Roofing Filter (Indicator)

variables:
        alpha1( 0 ),
        HP( 0 ),
        a1( 0 ),
        b1( 0 ),
        c1( 0 ),
        c2( 0 ),
        c3( 0 ),
        Filt( 0 ) ;

alpha1 = ( Cosine( .707 * 360 / 48 )
        + Sine ( .707 * 360 / 48 ) - 1 )
        / Cosine( .707 * 360 / 48 ) ;

HP = ( 1 - alpha1 / 2 ) * ( 1 - alpha1 / 2 )
        * ( Close - 2 * Close[1] + Close[2] )
        + 2 * ( 1 - alpha1 ) * HP[1] - ( 1 - alpha1 )
        * ( 1 - alpha1 ) * HP[2] ;

a1 = ExpValue( -1.414 * 3.14159 / 10 ) ;
b1 = 2 * a1 * Cosine( 1.414 * 180 / 10 ) ;
c2 = b1 ;
c3 = -a1 * a1 ;
c1 = 1 - c2 - c3 ;
Filt = c1 *( HP + HP[1] ) / 2
        + c2 * Filt[1] + c3 * Filt[2] ;

Plot1( Filt, "SS Filter" ) ;
Plot2( 0, "Zero Line" ) ;
```

_Ehlers_SuperSmoothStoch (Indicator)

```
inputs:
        Length( 20 ),
        OBLevel( .8 ),
        OSLevel( .2 ) ;

variables:
        alpha1( 0 ),
        HP( 0 ),
        a1( 0 ),
        b1( 0 ),
        c1( 0 ),
        c2( 0 ),
        c3( 0 ),
        Filt( 0 ),
        HighestC( 0 ),
        LowestC( 0 ),
        count( 0 ),
        Stoc( 0 ),
        MyStochastic( 0 ) ;

alpha1 = ( Cosine( .707 * 360 / 48 )
        + Sine ( .707 * 360 / 48 ) - 1 )
        / Cosine( .707 * 360 / 48 ) ;

HP = ( 1 - alpha1 / 2 ) * ( 1 - alpha1 / 2 )
        * ( Close - 2 * Close[1] + Close[2] )
        + 2 * ( 1 - alpha1 ) * HP[1] - ( 1 - alpha1 )
        * ( 1 - alpha1 ) * HP[2] ;

a1 = ExpValue( -1.414 * 3.14159 / 10 ) ;
b1 = 2 * a1 * Cosine( 1.414 * 180 / 10 ) ;
c2 = b1 ;
c3 = -a1 * a1 ;
c1 = 1 - c2 - c3 ;
Filt = c1 *( HP + HP[1] ) / 2
        + c2 * Filt[1] + c3 * Filt[2] ;

HighestC = Filt ;
LowestC = Filt ;

For count = 0 to Length - 1
        begin
        if Filt[count] > HighestC then
                HighestC = Filt[count] ;
        if Filt[count] < LowestC then
                LowestC = Filt[count] ;
        end;

Stoc = ( Filt - LowestC ) / ( HighestC - LowestC ) ;
MyStochastic = c1 * ( Stoc + Stoc[1] ) / 2 +
        c2 * MyStochastic[1] + c3 * MyStochastic[2] ;

Plot1( MyStochastic, "Stoch" ) ;
Plot2( OBLevel, "OverBought" ) ;
Plot6( OSLevel, "OverSold" ) ;
```

_Ehlers_SuperSmoothStoch (Strategy)

```
inputs:
        Length( 20 ),
        OBLevel( .8 ),
        OSLevel( .2 ) ;

variables:
        alpha1( 0 ),
        HP( 0 ),
        a1( 0 ),
        b1( 0 ),
        c1( 0 ),
        c2( 0 ),
        c3( 0 ),
        Filt( 0 ),
        HighestC( 0 ),
        LowestC( 0 ),
        count( 0 ),
        Stoc( 0 ),
        MyStochastic( 0 ) ;

alpha1 = ( Cosine( .707 * 360 / 48 )
        + Sine ( .707 * 360 / 48 ) - 1 )
        / Cosine( .707 * 360 / 48 ) ;

HP = ( 1 - alpha1 / 2 ) * ( 1 - alpha1 / 2 )
        * ( Close - 2 * Close[1] + Close[2] )
        + 2 * ( 1 - alpha1 ) * HP[1] - ( 1 - alpha1 )
        * ( 1 - alpha1 ) * HP[2] ;

a1 = ExpValue( -1.414 * 3.14159 / 10 ) ;
b1 = 2 * a1 * Cosine( 1.414 * 180 / 10 ) ;
c2 = b1 ;
c3 = -a1 * a1 ;
c1 = 1 - c2 - c3 ;
Filt = c1 *( HP + HP[1] ) / 2
        + c2 * Filt[1] + c3 * Filt[2] ;

HighestC = Filt ;
LowestC = Filt ;

For count = 0 to Length - 1
        begin
        if Filt[count] > HighestC then
                HighestC = Filt[count] ;
        if Filt[count] < LowestC then
                LowestC = Filt[count] ;
        end;

Stoc = ( Filt - LowestC ) / ( HighestC - LowestC ) ;
MyStochastic = c1 * ( Stoc + Stoc[1] ) / 2 +
        c2 * MyStochastic[1] + c3 * MyStochastic[2] ;

if MyStochastic crosses over ObLevel then
        SellShort next bar at Market
else if MyStochastic crosses under OsLevel then
        Buy next bar at Market ;
```

A sample chart is shown in Figure 1.

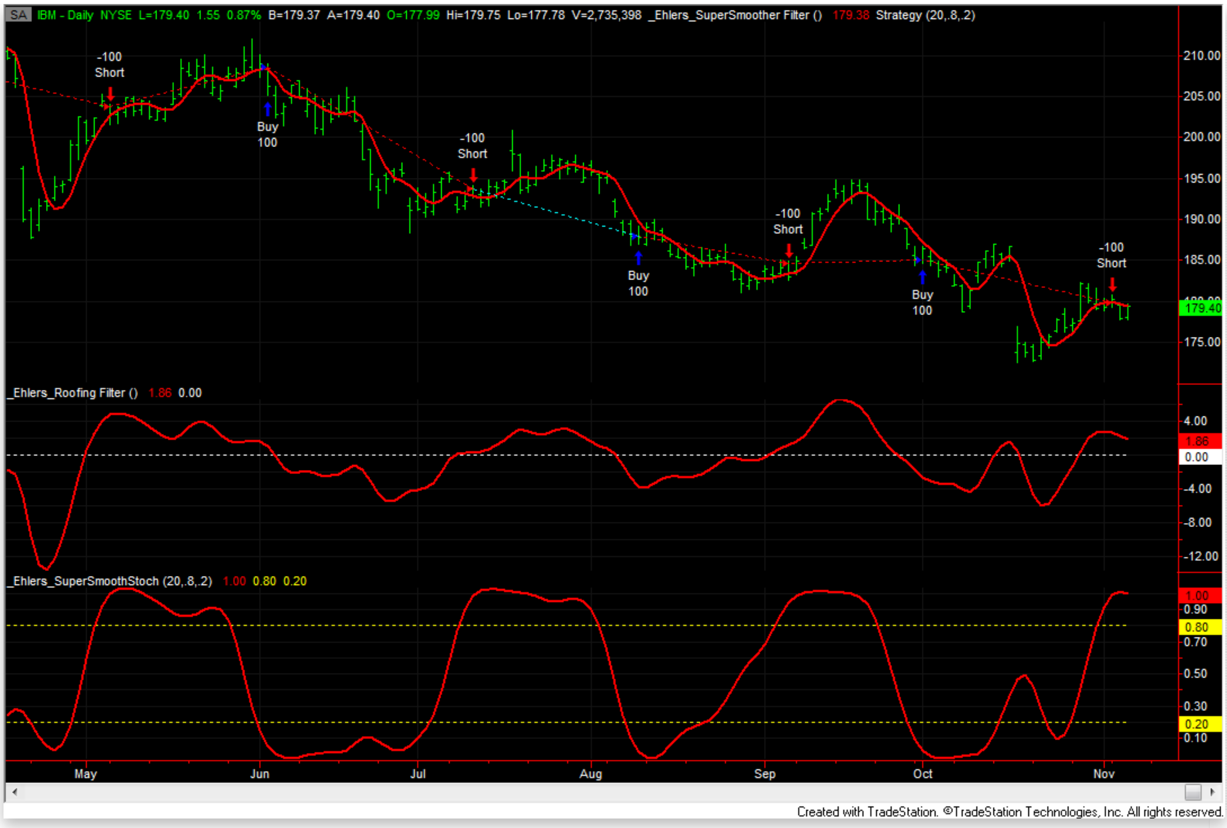


FIGURE 1: TRADESTATION. This daily chart of IBM shows the indicators and strategy from John Ehlers' article in this issue.

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.

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

BACK TO LIST

eSignal

eSIGNAL: JANUARY 2014 TRADERS' TIPS CODE

For this month's Traders' Tip, we've provided the formulas MESA_Stochastic_Indicator.efs, Roofing_Filter.efs, and SuperSmoother_Filter.efs, based on the formulas described in John Ehlers' article in this isuse, "Predictive And Successful Indicators."

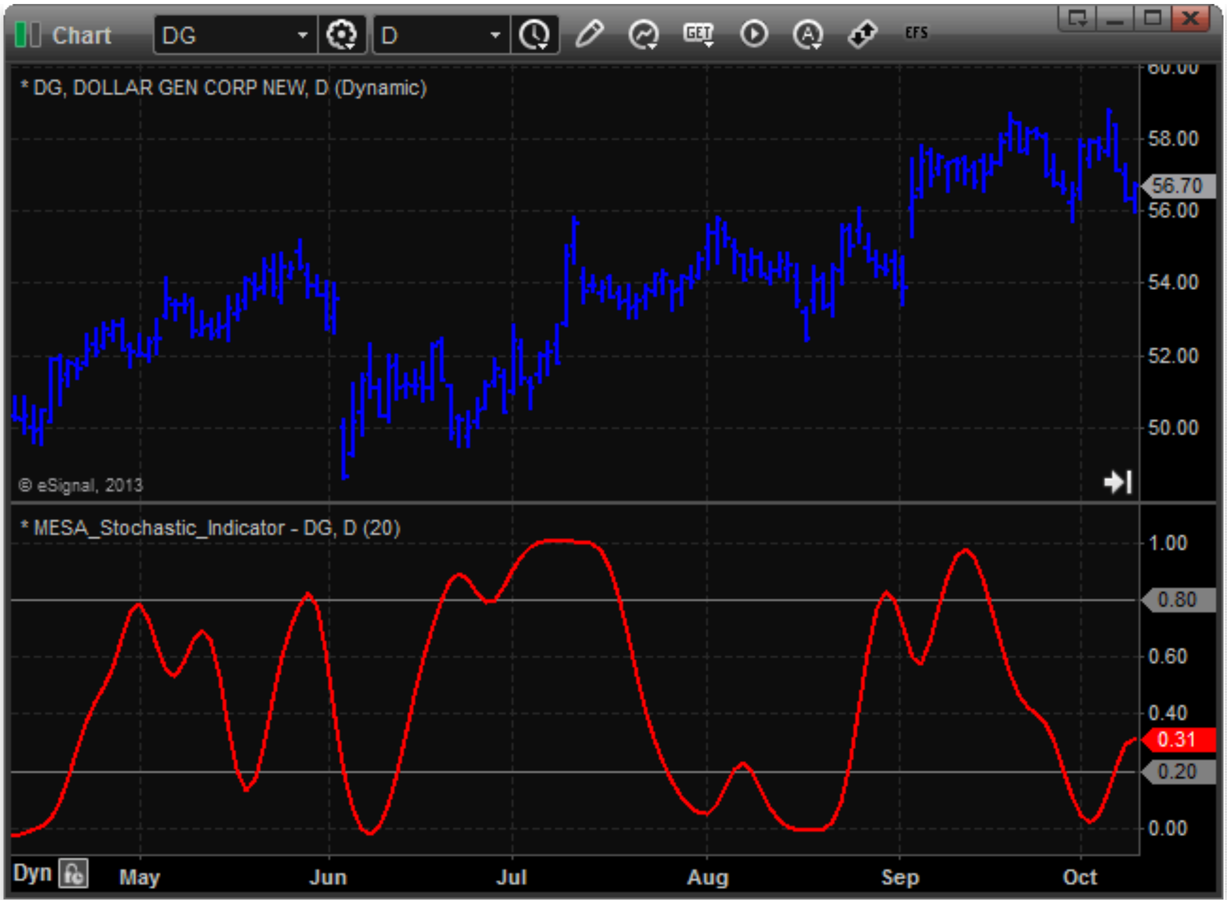


FIGURE 2: eSIGNAL, MESA STOCHASTIC

The MESA_Stochastic_Indicator study (Figure 2) contains one formula parameter, *length*, which may be configured through the *edit chart* window (right-click on *chart* and *select edit chart*). The roofing filter is shown in Figure 3.

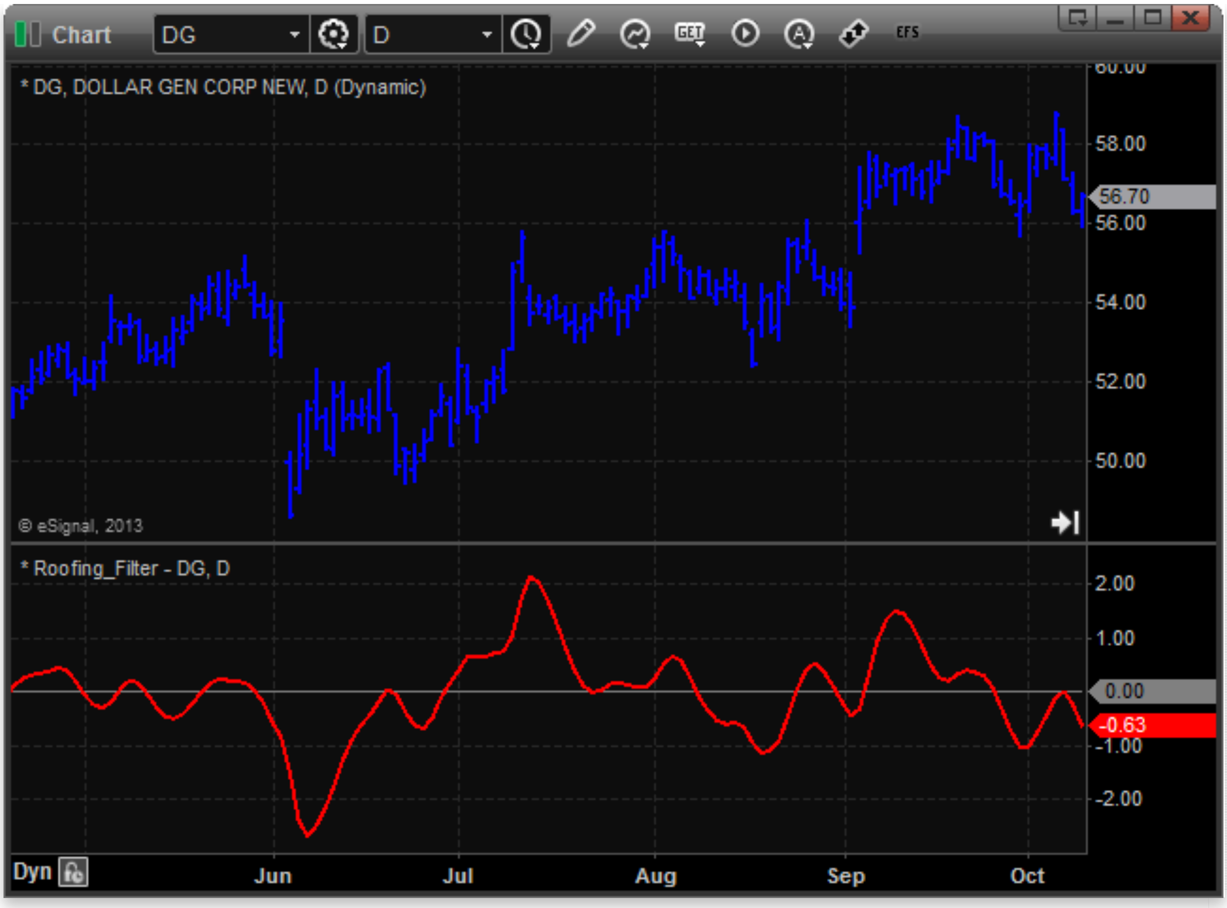


FIGURE 3: eSIGNAL, ROOFING FILTER

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 scripts (EFS) are also available for copying & pasting below.

MESA_Stochastic_Indicator.efs

```
/*********************************
Provided By:
    Interactive Data Corporation (Copyright Â© 2013)
    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:
    Stochastic Indicator by John F. Ehlers

Formula Parameters:                     Default:
Length                                  20

Version:            1.00  11/07/2013

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("MESA_Stochastic_Indicator");

    setCursorLabelName("BandLine1", 0);
    setCursorLabelName("BandLine2", 1);
    setCursorLabelName("MyStochastic", 2);

    setDefaultBarFgColor(Color.grey, 0);
    setDefaultBarFgColor(Color.grey, 1);
    setDefaultBarFgColor(Color.red, 2);

    setDefaultBarThickness(1, 0);
    setDefaultBarThickness(1, 1);
    setDefaultBarThickness(2, 2);

    setShowCursorLabel(false, 0);
    setShowCursorLabel(false, 1);
    setShowCursorLabel(true, 2);

    var x = 0;

    fpArray[x] = new FunctionParameter("fpLength", FunctionParameter.NUMBER);
    with(fpArray[x++])
    {
        setName("Length");
        setLowerLimit(2);
        setDefault(20);
    }
}

var bInit = false;
var bVersion = null;

var xHP = null;
var xFilt = null;
var xStoc = null;
var xMyStochastic = null;

var nBandLine1 = 0;
var nBandLine2 = 0;

function main(fpLength)
{
    if (bVersion == null) bVersion = verify();
    if (bVersion == false) return;

    if (!bInit)
    {
        xHP = efsInternal("Calc_HPSeries");
        xFilt = efsInternal("Calc_FiltSeries", xHP);
        xStoc = efsInternal("Calc_Stoc", xFilt, fpLength);
        xMyStochastic = efsInternal("Calc_MyStochastic", xStoc);
        nBandLine1 = 0.8;
        nBandLine2 = 0.2;

        bInit = true;
    }

    var nMyStochastic = xMyStochastic.getValue(0);

    return [nBandLine1, nBandLine2, nMyStochastic];
}

var xClose = null;

var alpha1 = 0;

function Calc_HPSeries()
{
    if (getBarState() == BARSTATE_ALLBARS)
    {
        xClose = close();

        alpha1 = (Math.cos((.707*360 / 48) * Math.PI / 180) +
                  Math.sin((.707*360 / 48) * Math.PI / 180) - 1) /
                  Math.cos((.707*360 / 48) * Math.PI / 180);
    }

    var nClose_0 = xClose.getValue(0);
    var nClose_1 = xClose.getValue(-1);
    var nClose_2 = xClose.getValue(-2);

    if (nClose_1 == null || nClose_2 == null)
        return;

    var nHP_1 = ref(-1);
    var nHP_2 = ref(-2);

    var nReturnValue = (1 - alpha1 / 2) * (1 - alpha1 / 2) * (nClose_0 - 2 * nClose_1
+ nClose_2)
                       + 2 * (1 - alpha1) * nHP_1 - (1 - alpha1) * (1 - alpha1) *
nHP_2;

    return nReturnValue;
}

var a1 = 0;
var b1 = 0;
var c1 = 0;
var c2 = 0;
var c3 = 0;

function Calc_FiltSeries(xHP)
{
    if (getBarState() == BARSTATE_ALLBARS)
    {
        a1 = Math.exp(-1.414 * 3.14159 / 10);
        b1 = 2 * a1 * Math.cos((1.414 * 180 / 10)* Math.PI / 180);
        c2 = b1;
        c3 = -a1 * a1;
        c1 = 1 - c2 - c3;
    }
    var nHP_0 = xHP.getValue(0);
    var nHP_1 = xHP.getValue(-1);

    if (nHP_0 == null || nHP_1 == null)
        return;

    var nFilt_1 = ref(-1);
    var nFilt_2 = ref(-2);

    var nReturnValue = c1 * (nHP_0 + nHP_1) / 2 + c2 * nFilt_1 + c3 * nFilt_2;

    return nReturnValue;
}

function Calc_Stoc(xFilt, nLength)
{
    var nFilt = xFilt.getValue(0);
    var nHighestC = xFilt.getValue(0);
    var nLowestC = xFilt.getValue(0);

    if (nHighestC == null || nLowestC == null || nFilt == null)
        return;

    for (var i = 0; i < nLength; i++)
    {
       nFilt_i = xFilt.getValue(-i);

       if (nFilt_i == null)
           return;

       if (nFilt_i > nHighestC)
           nHighestC =  nFilt_i;
       if (nFilt_i < nLowestC)
           nLowestC =  nFilt_i;
    }

    var nReturnValue = (nFilt - nLowestC) / (nHighestC - nLowestC);

    return nReturnValue;
}

function Calc_MyStochastic(xStoc)
{
    if (getBarState() == BARSTATE_ALLBARS)
    {
        a1 = Math.exp(-1.414 * 3.14159 / 10);
        b1 = 2 * a1 * Math.cos((1.414 * 180 / 10)* Math.PI / 180);
        c2 = b1;
        c3 = -a1 * a1;
        c1 = 1 - c2 - c3;
    }

    var nStoc_0 = xStoc.getValue(0);
    var nStoc_1 = xStoc.getValue(-1);

    if (nStoc_0 == null || nStoc_1 == null)
        return;

    var nMyStochastic_1 = ref(-1);
    var nMyStochastic_2 = ref(-2);

    var nReturnValue = c1 * (nStoc_0 + nStoc_1) / 2 + c2 * nMyStochastic_1 + c3 *
nMyStochastic_2;

    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;
}
```

Roofing_Filter.efs

```
/*********************************
Provided By:
    Interactive Data Corporation (Copyright © 2013)
    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:
    Roofing Filter by John F. Ehlers

Version:            1.00  11/07/2013

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

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

function preMain()
{
    setStudyTitle("Roofing_Filter");

    setCursorLabelName("ZeroLine", 0);
    setCursorLabelName("Roofing_Filter", 1);

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

    setDefaultBarThickness(1, 0);
    setDefaultBarThickness(2, 1);

    setShowCursorLabel(false, 0);
    setShowCursorLabel(true, 1);
}

var bInit = false;
var bVersion = null;

var xHP = null;
var xFilt = null;

function main()
{
    if (bVersion == null) bVersion = verify();
    if (bVersion == false) return;

    if (!bInit)
    {
        xHP = efsInternal("Calc_HPSeries");
        xFilt = efsInternal("Calc_FiltSeries", xHP);

        bInit = true;
    }

    var nZeroLine = 0;

    var nFilt = xFilt.getValue(0);

    return [nZeroLine, nFilt];
}

var xClose = null;

var alpha1 = 0;

function Calc_HPSeries()
{
    if (getBarState() == BARSTATE_ALLBARS)
    {
        xClose = close();

        alpha1 = (Math.cos((.707*360 / 48) * Math.PI / 180) +
                  Math.sin((.707*360 / 48) * Math.PI / 180) - 1) /
                  Math.cos((.707*360 / 48) * Math.PI / 180);
    }

    var nClose_0 = xClose.getValue(0);
    var nClose_1 = xClose.getValue(-1);
    var nClose_2 = xClose.getValue(-2);

    if (nClose_1 == null || nClose_2 == null)
        return;
    var nHP_1 = ref(-1);
    var nHP_2 = ref(-2);

    var nReturnValue = (1 - alpha1 / 2) * (1 - alpha1 / 2) * (nClose_0 - 2 * nClose_1
+ nClose_2)
                       + 2 * (1 - alpha1) * nHP_1 - (1 - alpha1) * (1 - alpha1) *
nHP_2;

    return nReturnValue;
}

var a1 = 0;
var b1 = 0;
var c1 = 0;
var c2 = 0;
var c3 = 0;

function Calc_FiltSeries(xHP)
{
    if (getBarState() == BARSTATE_ALLBARS)
    {
        a1 = Math.exp(-1.414 * 3.14159 / 10);
        b1 = 2 * a1 * Math.cos((1.414 * 180 / 10)* Math.PI / 180);
        c2 = b1;
        c3 = -a1 * a1;
        c1 = 1 - c2 - c3;
    }

    var nHP_0 = xHP.getValue(0);
    var nHP_1 = xHP.getValue(-1);

    if (nHP_0 == null || nHP_1 == null)
        return;

    var nFilt_1 = ref(-1);
    var nFilt_2 = ref(-2);

    var nReturnValue = c1 * (nHP_0 + nHP_1) / 2 + c2 * nFilt_1 + c3 * nFilt_2;

    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;
}
```

SuperSmoother_Filter.efs

```
/*********************************
Provided By:
    Interactive Data Corporation (Copyright © 2013)
    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 Super Smoother Filter by John F. Ehlers

Version:            1.00  11/07/2013

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

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

function preMain()
{
    setStudyTitle("SuperSmoother_Filter");
}

var bInit = false;
var bVersion = null;

var xFilt = null;

function main()
{
    if (bVersion == null) bVersion = verify();
    if (bVersion == false) return;

    if (!bInit)
    {
        xFilt = efsInternal("Calc_FiltSeries");

        bInit = true;
    }

    var nFilt = xFilt.getValue(0);

    return nFilt;
}

var xClose = null;

var a1 = 0;
var b1 = 0;
var c1 = 0;
var c2 = 0;
var c3 = 0;

function Calc_FiltSeries()
{
    if (getBarState() == BARSTATE_ALLBARS)
    {
        xClose = close();

        a1 = Math.exp(-1.414 * 3.14159 / 10);
        b1 = 2 * a1 * Math.cos((1.414 * 180 / 10) * Math.PI / 180);
        c2 = b1;
        c3 = -a1 * a1;
        c1 = 1 - c2 - c3;
    }

    var nClose_0 = xClose.getValue(0);
    var nClose_1 = xClose.getValue(-1);

    if (nClose_1 == null)
        return;

    var nFilt_1 = ref(-1);
    var nFilt_2 = ref(-2);

    var nReturnValue = c1 * (nClose_0 + nClose_1) / 2 + c2 * nFilt_1 + c3 * nFilt_2;

    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;
}
```

—Jason Keck
eSignal, an Interactive Data company
800 779-6555, www.eSignal.com

BACK TO LIST

THINKORSWIM: JANUARY 2014 TRADERS’ TIPS CODE

In “Predictive And Successful Indicators” in this issue, author John Ehlers presents another innovative way to eliminate noise from classic indicators and introduces some new smoothing indicators. For thinkorswim users, we have created three new studies and a strategy in our proprietary scripting language, thinkScript. You can adjust the parameters within the *edit studies* window to fine-tune your variables.

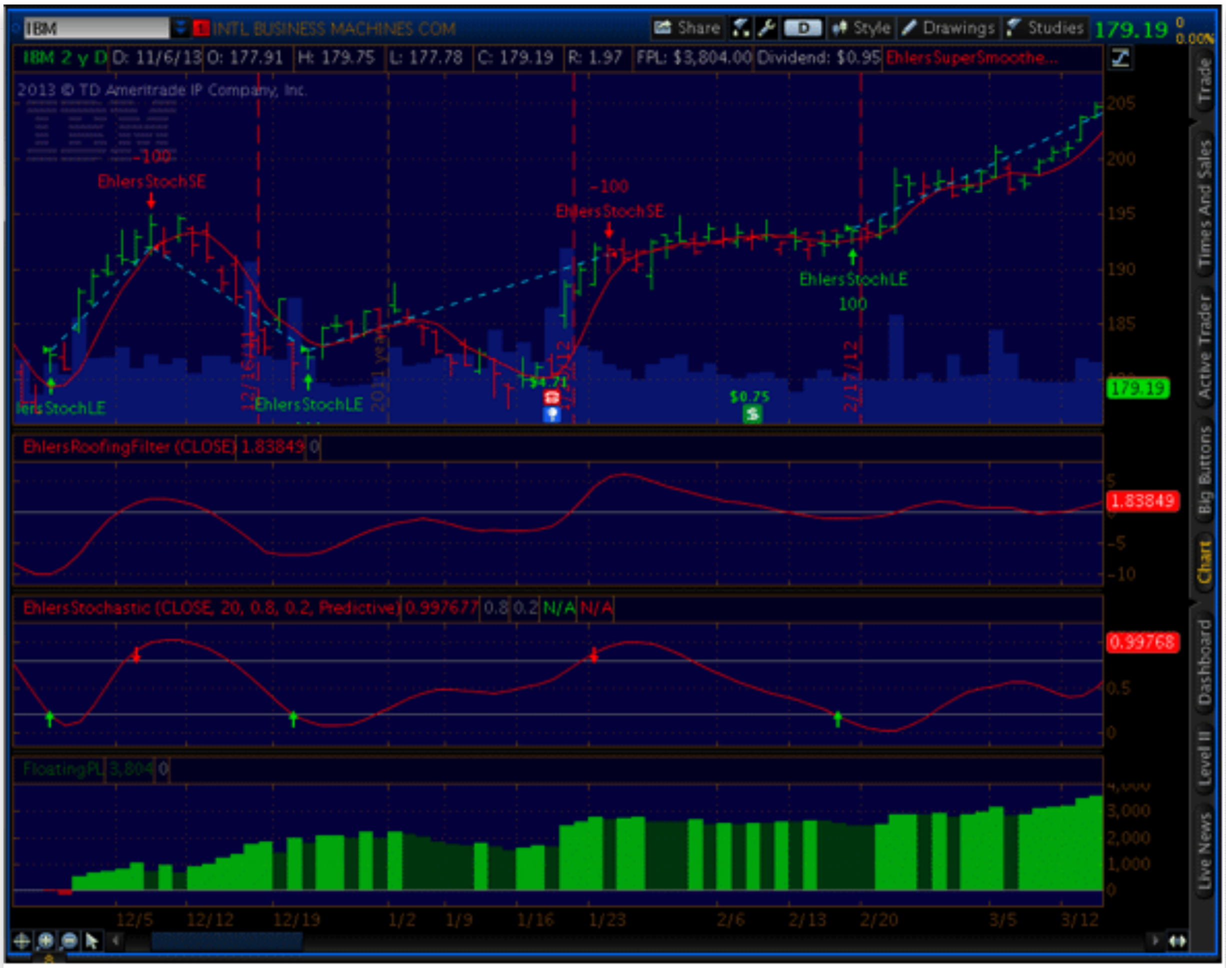


FIGURE 4: THINKORSWIM. A sample chart of IBM displays the roofing indicator and John Ehlers’ MESA Stochastic indicator described in his article in this issue.

Thinkorswim users can find the formulas for each indicator below:

- Strategy: http://tos.mx/Jiea7j

1. From our TOS Charts, Select “**Studies**” → “Edit **Studies**”.
2. Select the “**Strategy**” tab in the upper left hand corner.
3. Select “**New**” in the lower left hand corner.
4. Name the strategy (i.e. EhlersStoch)
5. Click in the script editor window, remove “**addOrder(OrderType.BUY_AUTO, no);**” and paste the following:

```
script EhlersSuperSmootherFilter {
    input price = close;

    def a1 = Exp(-Double.Pi * Sqrt(2) / 10);
    def coeff2 = 2 * a1 * Cos(Sqrt(2) * Double.Pi / 10);
    def coeff3 = - Sqr(a1);
    def coeff1 = 1 - coeff2 - coeff3;
    def filt = coeff1 * (price + price[1]) / 2 + coeff2 * filt[1] + coeff3
* filt[2];

    plot SuperSmootherFilter = filt;
}

script EhlersRoofingFilter {
    input price = close;

    def alpha1 = (Cos(Sqrt(2) * Double.Pi / 48) + Sin (Sqrt(2) * Double.Pi
/ 48) - 1) / Cos(Sqrt(2) * Double.Pi / 48);
    def highpass = Sqr(1 - alpha1 / 2) * (price - 2 * price[1] + price[2])
+ 2 * (1 - alpha1) * highpass[1] - Sqr(1 - alpha1) * highpass[2];

    plot RoofingFilter = reference EhlersSuperSmootherFilter(highpass);
}

script EhlersStochastic {
    input price = close;
    input length = 20;
    input over_bought = .8;
    input over_sold = .2;
    input mode = {default Predictive, Conventional};

    def filt = reference EhlersRoofingFilter(price);
    def highestP = Highest(filt, length);
    def lowestP = Lowest(filt, length);
    def stoch = if (highestP - lowestP) != 0 then (filt - lowestP) /
(highestP - lowestP) else 0;

    plot Stochastic = reference EhlersSuperSmootherFilter(stoch);
    plot OverBought = over_bought;
    plot OverSold = over_sold;
    plot Buy;
    plot Sell;
    switch (mode) {
    case Predictive:
        Buy = if Stochastic crosses below OverSold then OverSold + .05 else
Double.NaN;
        Sell = if Stochastic crosses above OverBought then OverBought - .05
else Double.NaN;
    case Conventional:
        Buy = if Stochastic crosses above OverSold then OverSold + .05 else
Double.NaN;
        Sell = if Stochastic crosses below OverBought then OverBought - .05
else Double.NaN;
    }
}

input price = close;
input length = 20;
input over_bought = .8;
input over_sold = .2;
input mode = {default Predictive, Conventional};

def buy = reference EhlersStochastic(price, length, over_bought, over_sold,
mode).Buy;
def sell = reference EhlersStochastic(price, length, over_bought,
over_sold, mode).Sell;

AddOrder(OrderType.BUY_AUTO, !IsNaN(buy), tickColor = Color.UPTICK,
arrowColor = Color.UPTICK, name = "EhlersStochLE");
AddOrder(OrderType.SELL_AUTO, !IsNaN(sell), tickColor = Color.DOWNTICK,
arrowColor = Color.DOWNTICK, name = "EhlersStochSE");
```

- EhlersSuperSmootherFilter: http://tos.mx/uRNvch

  1. From our TOS Charts, Select “**Studies**” → “**Edit Studies**”.
  2. Select the “**Strategies**” tab in the upper left hand corner.
  3. Select “**New**” in the lower left hand corner.
  4. Name the study (i.e. EhlersSuperSmootherFilter)
  5. Click in the script editor window, remove “**plot data = close;**” and paste the following:

```
input price = close;

def a1 = Exp(-Double.Pi * Sqrt(2) / 10);
def coeff2 = 2 * a1 * Cos(Sqrt(2) * Double.Pi / 10);
def coeff3 = - Sqr(a1);
def coeff1 = 1 - coeff2 - coeff3;
def filt = coeff1 * (price + price[1]) / 2 + coeff2 * filt[1] + coeff3 *
filt[2];

plot SuperSmootherFilter = filt;
SuperSmootherFilter.SetDefaultColor(GetColor(5));
```

- EhlersRoofingFilter study: http://tos.mx/dcQPdE

  1. From our TOS Charts, Select “**Studies**” → “**Edit Studies**”.
  2. Select the “**Strategies**” tab in the upper left hand corner.
  3. Select “**New**” in the lower left hand corner.

4. Name the study (i.e. EhlersRoofingFilter)
5. Click in the script editor window, remove “**plot data = close;**” and paste the following:

```
script EhlersSuperSmootherFilter {
    input price = close;

    def a1 = Exp(-Double.Pi * Sqrt(2) / 10);
    def coeff2 = 2 * a1 * Cos(Sqrt(2) * Double.Pi / 10);
    def coeff3 = - Sqr(a1);
    def coeff1 = 1 - coeff2 - coeff3;
    def filt = coeff1 * (price + price[1]) / 2 + coeff2 * filt[1] + coeff3
* filt[2];

    plot SuperSmootherFilter = filt;
}

declare lower;

input price = close;

def alpha1 = (Cos(Sqrt(2) * Double.Pi / 48) + Sin (Sqrt(2) * Double.Pi /
48) - 1) / Cos(Sqrt(2) * Double.Pi / 48);
def highpass = Sqr(1 - alpha1 / 2) * (price - 2 * price[1] + price[2]) + 2
* (1 - alpha1) * highpass[1] - Sqr(1 - alpha1) * highpass[2];

plot RoofingFilter = reference EhlersSuperSmootherFilter(highpass);
plot ZeroLine = 0;

RoofingFilter.SetDefaultColor(GetColor(5));
ZeroLine.SetDefaultColor(GetColor(7));
```

- EhlersStochastic study: http://tos.mx/yVruCn

1. From our TOS Charts, Select “**Studies**” → “**Edit Studies**”.
2. Select the “**Strategies**” tab in the upper left hand corner.
3. Select “**New**” in the lower left hand corner.
4. Name the study (i.e. EhlersStochastic)
5. Click in the script editor window, remove “**plot data = close;**” and paste the following:

```
script EhlersSuperSmootherFilter {
    input price = close;

    def a1 = Exp(-Double.Pi * Sqrt(2) / 10);
    def coeff2 = 2 * a1 * Cos(Sqrt(2) * Double.Pi / 10);
    def coeff3 = - Sqr(a1);
    def coeff1 = 1 - coeff2 - coeff3;
    def filt = coeff1 * (price + price[1]) / 2 + coeff2 * filt[1] + coeff3
* filt[2];

    plot SuperSmootherFilter = filt;
}
```

```
script EhlersRoofingFilter {
    input price = close;

    def alpha1 = (Cos(Sqrt(2) * Double.Pi / 48) + Sin (Sqrt(2) * Double.Pi
/ 48) - 1) / Cos(Sqrt(2) * Double.Pi / 48);
    def highpass = Sqr(1 - alpha1 / 2) * (price - 2 * price[1] + price[2])
+ 2 * (1 - alpha1) * highpass[1] - Sqr(1 - alpha1) * highpass[2];

    plot RoofingFilter = reference EhlersSuperSmootherFilter(highpass);
}

declare lower;

input price = close;
input length = 20;
input over_bought = .8;
input over_sold = .2;
input mode = {default Predictive, Conventional};

def filt = reference EhlersRoofingFilter(price);
def highestP = Highest(filt, length);
def lowestP = Lowest(filt, length);
def stoch = if (highestP - lowestP) != 0 then (filt - lowestP) / (highestP
- lowestP) else 0;

plot Stochastic = reference EhlersSuperSmootherFilter(stoch);
plot OverBought = over_bought;
plot OverSold = over_sold;
plot Buy;
plot Sell;
switch (mode) {
case Predictive:
    Buy = if Stochastic crosses below OverSold then OverSold + .05 else
Double.NaN;
    Sell = if Stochastic crosses above OverBought then OverBought - .05
else Double.NaN;
case Conventional:
    Buy = if Stochastic crosses above OverSold then OverSold + .05 else
Double.NaN;
    Sell = if Stochastic crosses below OverBought then OverBought - .05
else Double.NaN;
}

Stochastic.SetDefaultColor(GetColor(5));
OverBought.SetDefaultColor(GetColor(7));
OverSold.SetDefaultColor(GetColor(7));
Buy.SetDefaultColor(Color.UPTICK);
Buy.SetPaintingStrategy(PaintingStrategy.ARROW_UP);
Buy.HideBubble();
Sell.SetDefaultColor(Color.DOWNTICK);
Sell.SetPaintingStrategy(PaintingStrategy.ARROW_DOWN);
Sell.HideBubble();
```

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

BACK TO

LIST

MetaStock

METASTOCK: JANUARY 2014 TRADERS’ TIPS CODE

John Ehlers’ article in this issue, “Predictive And Successful Indicators,” presents the reader with three new indicators. The MetaStock formulas for these indicators are given here:

```
SuperSmoother filter:
a1:= Exp(-1.414 * 3.14159 / 10);
b1:= 2*a1 * Cos(1.414*180 /10);
c2:= b1;
c3:= -a1 * a1;
c1:= 1 - c2 - c3;
c1 * (C + Ref(C, -1))/2 + c2*PREV + c3*Ref(PREV,-1)

Roofing filter:
alpha1:= (Cos(.707*360/48) + Sin(.707*360/48) -1)/Cos(.707*360/48);
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);
a1:= Exp(-1.414 * 3.14159 / 10);
b1:= 2*a1 * Cos(1.414*180 /10);
c2:= b1;
c3:= -a1 * a1;
c1:= 1 - c2 - c3;
c1 * (HP + Ref(HP, -1))/2 + c2*PREV + c3*Ref(PREV,-1)

MESA Stochastic:
alpha1:= (Cos(.707*360/48) + Sin(.707*360/48) -1)/Cos(.707*360/48);
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);
a1:= Exp(-1.414 * 3.14159 / 10);
b1:= 2*a1 * Cos(1.414*180 /10);
c2:= b1;
c3:= -a1 * a1;
c1:= 1 - c2 - c3;
filt:= c1 * (HP + Ref(HP, -1))/2 + c2*PREV + c3*Ref(PREV,-1);
stoc:= (filt-LLV(filt,20))/(HHV(filt,20)-LLV(filt,20));
c1 * (stoc + Ref(stoc, -1))/2 + c2*PREV + c3*Ref(PREV,-1);
```

—William Golson
MetaStock Technical Support
www.metastock.com

BACK TO LIST

WealthLab.com

WEALTH-LAB: JANUARY 2014 TRADERS’ TIPS CODE

In his article in this issue, “Predictive And Successful Indicators,” author John Ehlers presents two new indicators: the *SuperSmoother filter*, which is superior to moving averages for removing aliasing noise, and the *MESA Stochastic* oscillator, a stochastic successor that removes the effect of spectral dilation through the use of a roofing filter.

To demonstrate the effects of using the new indicators, Ehlers introduces a simple countertrend system that goes long when MESA Stochastic crosses below the oversold value and reverses the trade by taking a short position when the oscillator exceeds the overbought threshold.

To execute the trading system that Ehlers includes in his article, Wealth-Lab users need to install (or update) the latest version of our 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 5.

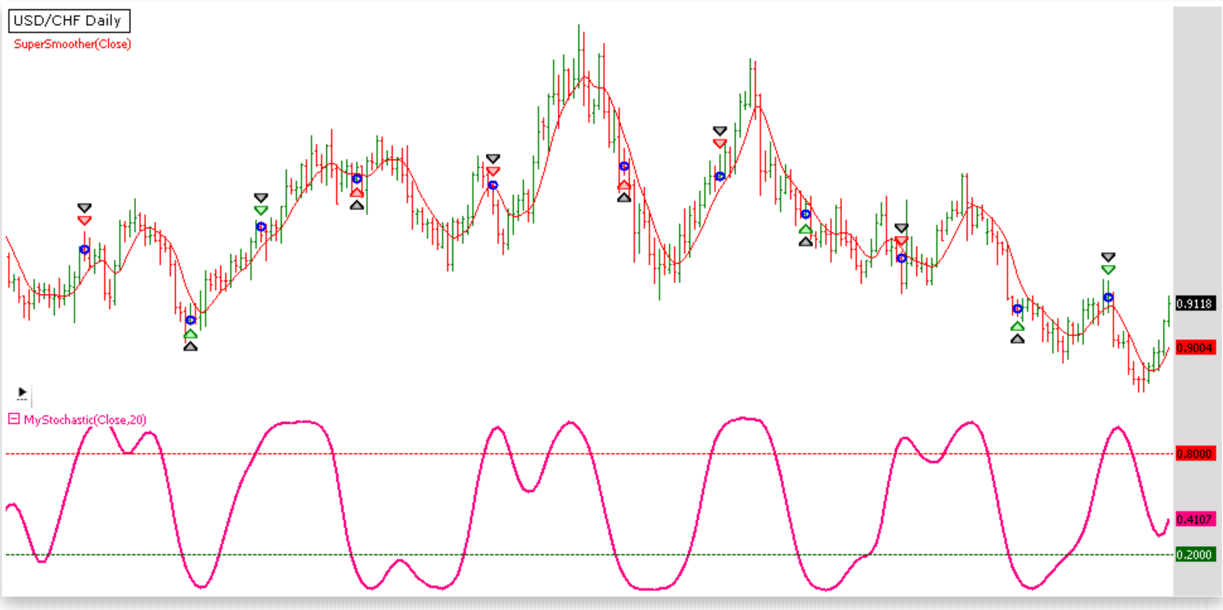


FIGURE 5: WEALTH-LAB. Here is a sample Wealth-Lab 6 chart illustrating application of the system’s rules on a daily chart of USD/CHF (US dollar to Swiss franc exchange rate).

```
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 Ehlers201401 : WealthScript
        {
                private StrategyParameter paramPeriod;

                public Ehlers201401()
                {
                        paramPeriod = CreateParameter("MESAStoch.Period",20,5,100,5);
                }

                protected override void Execute()
                {
                        int period = paramPeriod.ValueInt;
                        RoofingFilter rf = RoofingFilter.Series(Close);
                        MESAStochastic ms = MESAStochastic.Series(Close,period);

                        for(int bar = period; bar < Bars.Count; bar++)
                        {
                                // Detect crossover/crossunder and store state in a
variable
                                bool maXo = CrossOver(bar, ms, 0.8);
                                bool maXu = CrossUnder(bar, ms, 0.2);

                                // The first trade
                                if (Positions.Count == 0){
                                        if ( maXu )
                                                BuyAtMarket( bar + 1 );
                                        else if( maXo )
                                                ShortAtMarket( bar + 1 );
                                }
                                        // Subsequent trades
                                else
                                {
                                        Position p =
LastPosition;
                                        if ( p.PositionType == PositionType.Long )
                                        {
                                                if ( maXo )
                                                {
                                                        SellAtMarket( bar + 1, p );
                                                        ShortAtMarket( bar + 1 );
                                                }
                                        }
                                        else if ( maXu )
                                        {
                                                CoverAtMarket( bar + 1, p );
                                                BuyAtMarket( bar + 1 );
                                        }
                                }
                        }

                        HideVolume();
                        ChartPane pMS = CreatePane(40,false,true);
                        ChartPane pRF = CreatePane(20,true,true);

PlotSeries(PricePane,SuperSmoother.Series(Close),Color.Red,LineStyle.Solid,1);

PlotSeries(pRF,rf,Color.FromArgb(255,0,0,139),LineStyle.Solid,1);

PlotSeries(pMS,ms,Color.FromArgb(255,255,0,128),LineStyle.Solid,2);
                        DrawHorzLine(pMS,0.8,Color.Red,LineStyle.Dashed,1);
```

```
                                DrawHorzLine(pMS,0.2,Color.DarkGreen,LineStyle.Dashed,1);
                        }
                }
        }
```

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

BACK TO LIST

AMIBROKER: JANUARY 2014 TRADERS' TIPS CODE

In "Predictive And Successful Indicators" in this issue, author John Ehlers presents his SuperSmooth filter and uses it to create a better stochastic indicator.

Ready-to-use AmiBroker code for the indicator is shown below. Note that SuperSmooth filter and high-pass filter have been written as reusable, general-purpose functions so users can easily include them in their own systems and/or indicators. To display the indicators on a chart, simply input or paste the code into the formula editor and press *apply indicator*. To backtest a trading system, choose *backtest* from *Tools* menu in the formula editor.

A sample chart is shown in Figure 6.

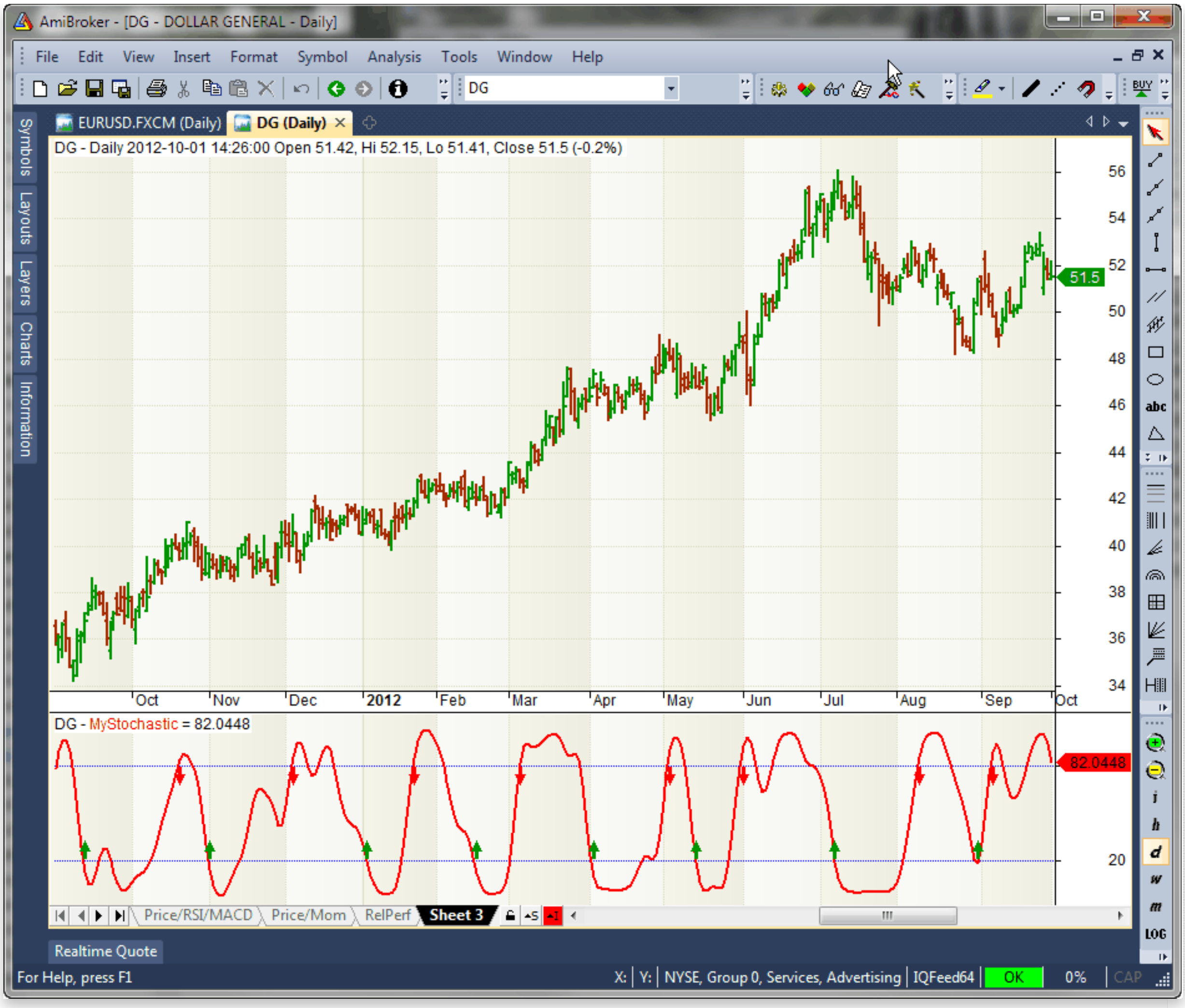

AmiBroker - [DG - DOLLAR GENERAL - Daily]
File Edit View Insert Format Symbol Analysis Tools Window Help
DG
Symbols
Layouts
Layers
Charts
Information
EURUSD.FXCM (Daily)
DG (Daily)
DG - Daily 2012-10-01 14:26:00 Open 51.42, Hi 52.15, Lo 51.41, Close 51.5 (-0.2%)
56
54
52
51.5
50
48
46
44
42
40
38
36
34
Oct
Nov
Dec
2012
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
DG - MyStochastic = 82.0448
82.0448
20
Price/RSI/MACD
Price/Mom
RelPerf
Sheet 3
Realtime Quote
For Help, press F1
X:
Y:
NYSE, Group 0, Services, Advertising
IQFeed64
OK
0%
CAP

FIGURE 6: AMIBROKER. Here is a sample price chart of DG with a SuperSmoothed stochastic indicator.

```
PI = 3.1415926;
SQ2 = sqrt( 2 );

function SuperSmoother( array, periods )
{
    a1 = exp( -SQ2 * PI / periods );
    b1 = 2 * a1 * cos( SQ2 * PI / periods );
    c2 = b1;
    c3 = -a1 * a1;
    c1 = 1 - c2 - c3;

    Filt = Nz( array );

    for ( i = 2; i < BarCount; i++ )
    {
         Filt[ i ] = c1 * ( array[ i ] + array[ i - 1 ] ) / 2 +
                     c2 * Filt[ i - 1 ] +
                     c3 * Filt[ i - 2];
    }
    return Filt;
}

function HighpassFilter( array, periods )
{
    alpha1 = ( cos( SQ2 * PI / periods ) + sin ( SQ2 * PI / periods ) - 1 ) /  cos(
SQ2 * PI / periods );

    HP = Nz( array );

    C1 = ( 1 - alpha1 / 2 ) ^ 2;
    C2 = 2 * ( 1 - alpha1 );
    C3 = - ( ( 1 - alpha1 ) ^ 2 );

    for ( i = 2; i < BarCount; i++ )
    {
        HP[ i ] =  C1 * ( array[ i ] - 2 * array[i-1] + array[i-2] ) +
                   C2 * HP[ i - 1] +
                   C3 * HP[ i - 2];
    }

    return HP;
}

ss = SuperSmoother( HighpassFilter( Close, 48 ), 10 );

Length = 20;


HighestC = HHV( ss, Length );
LowestC = LLV( ss, Length );

Stoc = ( ss - LowestC ) / ( HighestC - LowestC );

MyStochastic = 100 * SuperSmoother( Stoc, 10 );

Plot( MyStochastic, "MyStochastic", colorRed, styleThick );
PlotGrid( 20, colorBlue );
PlotGrid( 80, colorBlue );

Buy = Cross( 20, MyStochastic );
Sell = Cross( MyStochastic, 80 );

Buy = ExRem( Buy, Sell );
Sell = ExRem( Sell, Buy );
PlotShapes( Buy * shapeUpArrow, colorGreen, 0, 20, 8 );
PlotShapes( Sell * shapeDownArrow, colorRed, 0, 80, 8 );
```

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

BACK TO LIST

NEUROSHELL TRADER: JANUARY 2014 TRADERS' TIPS CODE

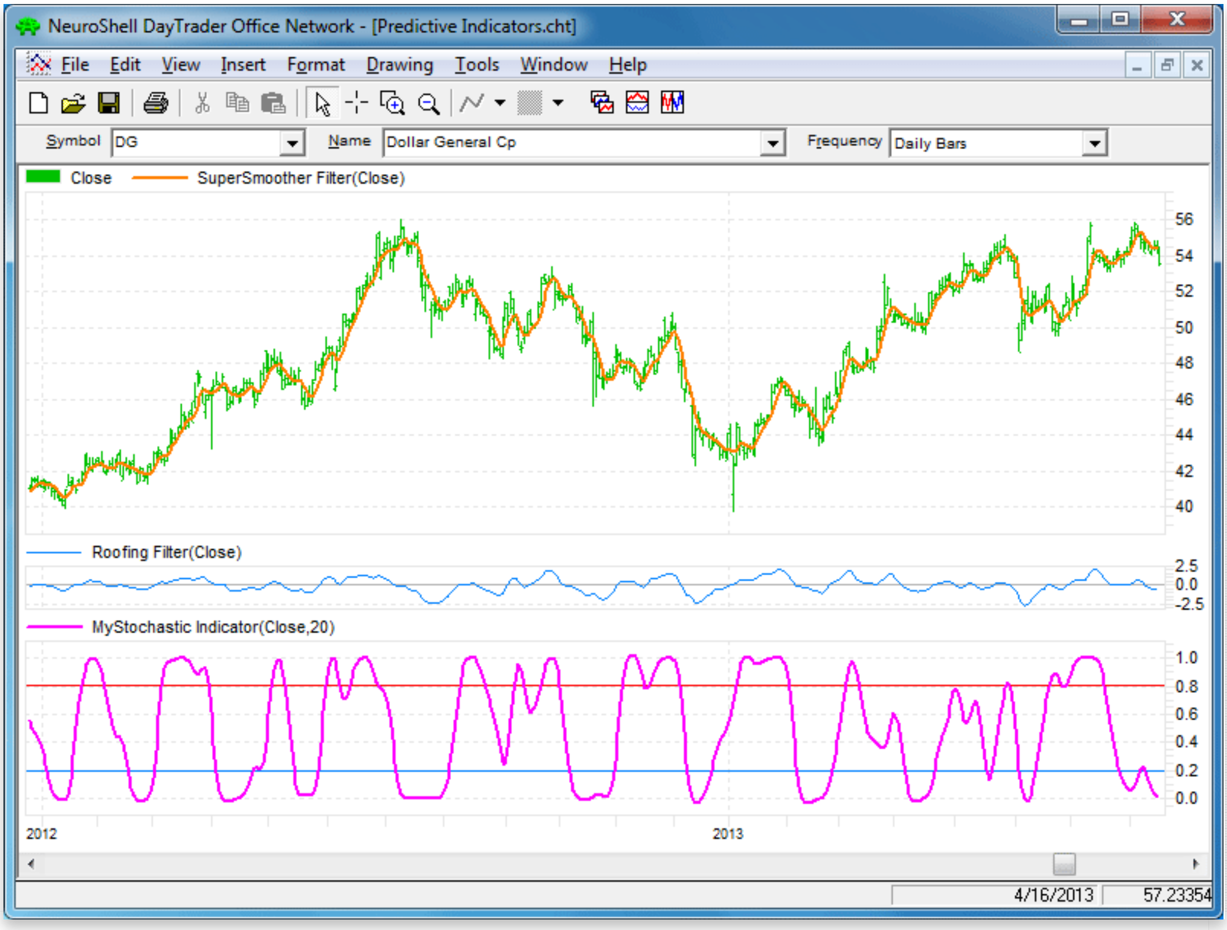
John Ehlers' SuperSmoother filter, roofing filter, and MESA Stochastic indicator presented in his article in this issue, "Predictive And Successful Indicators," can be easily implemented in NeuroShell Trader using NeuroShell Trader's ability to call external dynamic linked libraries. Dynamic linked libraries 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. Setup the parameters to match your DLL.
5. Select the **Finished** button.

Dynamic trading systems can be easily created in NeuroShell Trader by combining the MESA Stochastic indicator 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. 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.



Symbol DG
Name Dollar General Cp
Frequency Daily Bars

4/16/2013 57.23354

FIGURE 7: NEUROSHELL TRADER. This NeuroShell Trader chart displays John Ehlers' SuperSmoother filter, roofing filter, and MESA Stochastic indicator.

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

BACK TO LIST

AIQ: JANUARY 2014 TRADERS' TIPS CODE

This Traders' Tip is based on Ajay Pankhania's September 2013 article in STOCKS & COMMODITIES, "Muscle Up Those Averages."

The AIQ code for the moving averages and the MACD indicator discussed in the article is provided at www.TradersEdgeSystems.com/traderstips.htm.

This code may be useful to beginning AIQ Expert Design Studio users since it illustrates how to code multiple versions of the simple and the exponential moving averages as well as the MACD indicator. In addition, I have coded the simple moving average crossover system and the MACD crossover system.

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: JANUARY 2014 TRADERS' TIPS CODE

The TradersStudio code based on John Ehlers' article in this issue, "Predictive And Successful Indicators," 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: SUPSMO—The SuperSmoother filter returns a smoothed value for the input *price*
- Function: ROOF—The roofing filter returns a two-pole high-pass filtered value, which is then smoothed by the first function, the "SUPSMO" for the input *price*
- Function: MESASTOC—Calculates a stochastic of the *price* input that uses both the roofing filter and the SuperSmoother filters and returns a super-smoothed value for the stochastic
- Indicator: EHLERS_SUPSMO_IND is used to plot the SuperSmoother on a chart
- Indicator: EHLERS_ROOF_IND is used to plot the roofing filter on a chart
- Indicator: EHLERS_MESASTOC_IND is used to plot the MESA Stochastic on a chart
- System: EHLERS_MESASTOC_SYS is a system that I created (not found in Ehlers' article) to show an example of how to use the MESA Stochastic indicator in a system.

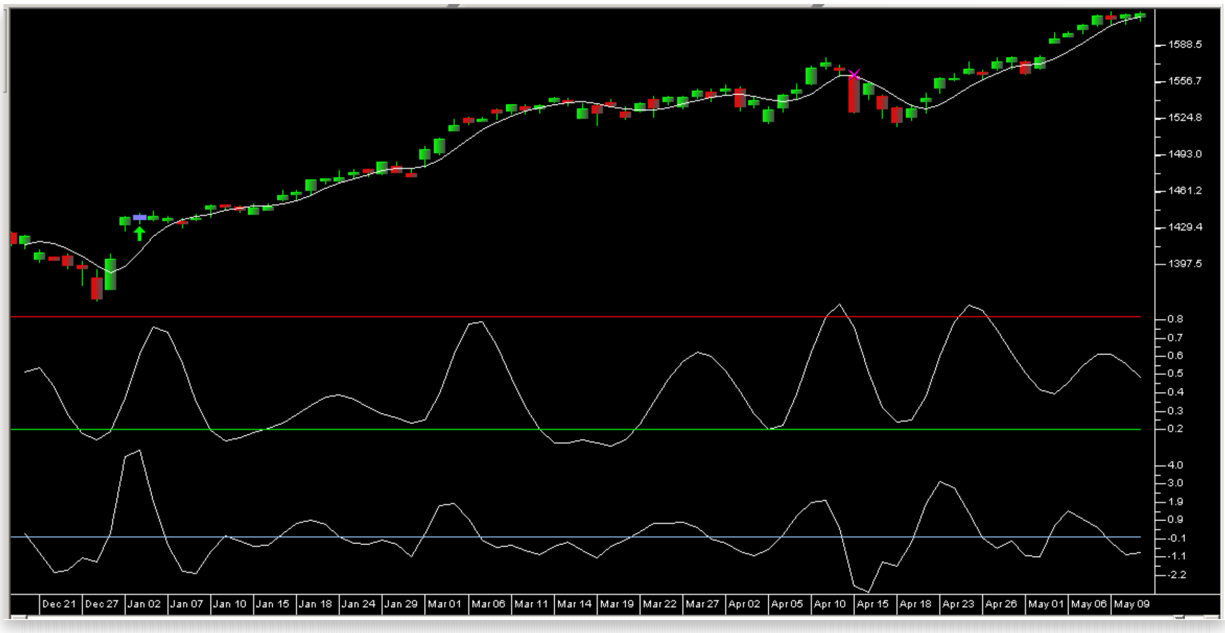


FIGURE 8: TRADERSSTUDIO, FUTURES CONTRACT. Here is a sample chart of the S&P 500 futures contract with the three indicators described by John Ehlers in his article in this issue, SuperSmoothing, roofing, and MESA Stochastic.

In Figure 8, I show a chart of the S&P 500 futures contract using Pinnacle Data, symbol “SP,” with the three indicators discussed by Ehlers in his article. In the main panel, we see the SuperSmoother filter using the close as the price input. In the second panel, we see the MESA Stochastic indicator, and in the bottom panel, we see the roofing filter. In Figure 9, I show the equity curve and underwater equity curve for the example system I wrote trading one contract, long only.

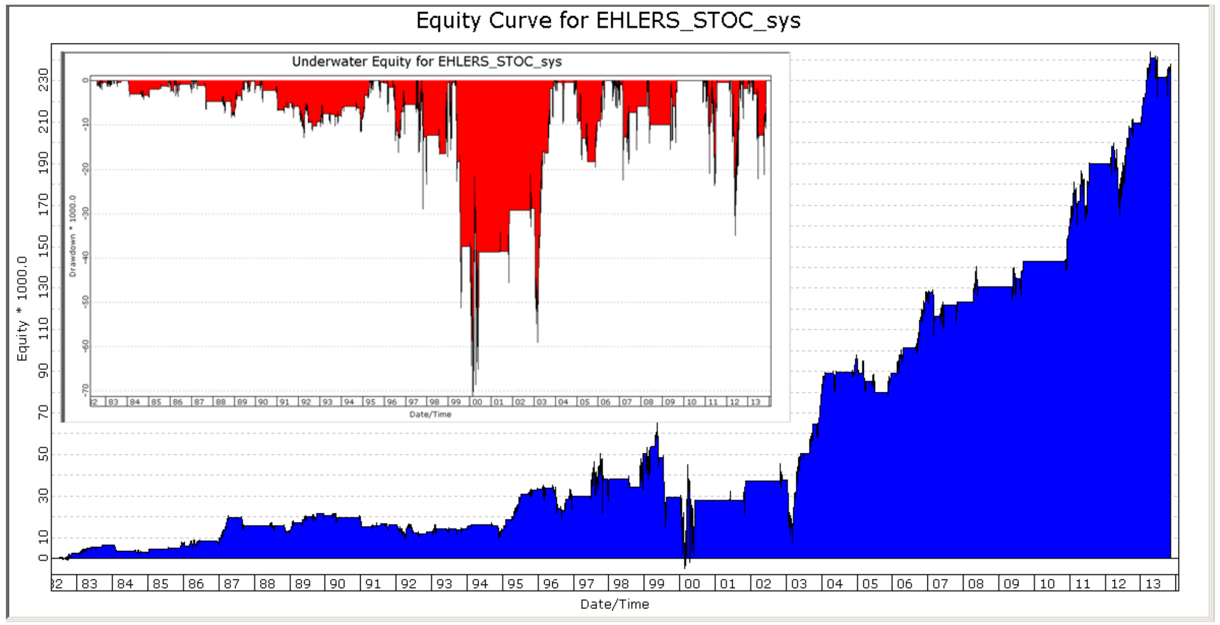


FIGURE 9: TRADERSSTUDIO, EQUITY CURVE. Here are equity and underwater curves for the example system I wrote trading one contract of the SP long-only.

```
'PREDICTIVE INDICATORS FOR EFFECTIVE TRADING STRATEGIES
'Author: John Ehlers, TASC January 2014
'Coded by: Richard Denning 11/02/2013
'www.TradersEdgeSystems.com

'SuperSmoother filter
' 2013 John F. Ehlers
Function SUPSMO(Price As BarArray)
Dim a1, b1, c1, c2, c3, Filt As BarArray
a1 = Exp(-1.414*3.14159 / 10)
b1 = 2*a1*Cos(DegToRad(1.414*180 / 10))
c2 = b1
c3 = -a1*a1
c1 = 1 - c2 - c3
Filt = c1*(Price + Price[1]) / 2 + c2*Filt[1] + c3*Filt[2]
SUPSMO = Filt
End Function
'-----------------------------------------------------------
'Roofing filter
' 2013 John F. Ehlers
Function ROOF(Price As BarArray)
Dim alpha1
Dim HP As BarArray
'Highpass filter cyclic components whose periods are shorter than 48 bars
  alpha1 = (Cos(DegToRad(.707*360 / 48) + Sin(DegToRad(.707*360 / 48) - 1)))
/Cos(DegToRad(.707*360 / 48))
  HP = (1 - alpha1 / 2)*(1 - alpha1 /2)*(Price - 2*Price[1] + Price[2]) + 2*(1 -
alpha1)*HP[1] - (1 - alpha1)*(1 - alpha1)*HP[2]
'Smooth with a Super Smoother Filter
  'assert(false)
  ROOF = SUPSMO(HP)
  End Function
'-----------------------------------------------------------
'MesaStochastic Indicator
' 2013 John F. Ehlers
Function MESASTOC(Price as bararray,stoLen)
  Dim count
  Dim HighestC As BarArray
  Dim LowestC As BarArray
  Dim stoc As BarArray
  Dim filtSmoPrice As BarArray
  Dim myStochastic As BarArray

'Highpass filter cyclic components whose periods are shorter than 48 bars
' then smooted with a super smother filter
  filtSmoPrice = ROOF(Price)
  HighestC = filtSmoPrice
  LowestC = filtSmoPrice
  For count = 0 To stoLen - 1
    If filtSmoPrice[count] > HighestC Then HighestC = filtSmoPrice[count]
    If filtSmoPrice[count] < LowestC Then LowestC = filtSmoPrice[count]
  Next
  stoc = (filtSmoPrice - LowestC) / (HighestC - LowestC)
'Smooth with a Super Smoother Filter
  myStochastic = SUPSMO(stoc)
MESASTOC = myStochastic
End Function
'----------------------------------------------------------
'Indicator plot for SuperSmoother Filter
Sub EHLERS_SUPSMO_IND()
Plot1(SUPSMO(Close))
End Sub
'-----------------------------------------------------------
'Indicator plot for Roofing Filter
Sub EHLERS_ROOF_IND()
plot1(ROOF(Close))
plot2(0)
End Sub
'------------------------------------------------------------
'MesaStochastic Indicator Plot
' 2013 John F. Ehlers
Sub EHLERS_MESASTOC_IND(stoLen)
  Plot1(MesaStoc(Close,stoLen))
  Plot2(.8)
  Plot3(.2)
End Sub
'-----------------------------------------------------------
'Example of how one micht use Ehler's MesaStochastic Indicator in a system (long only)
Sub EHLERS_MESASTOC_SYS(stoLen,upperVal,lowerVal,trndLen)
  'default parameter values: stoLen=20,upperVal=0.8,lowerVal=0.2,trndLen=25
  Dim smoFiltSto
  Dim smoFiltClose As BarArray
  smoFiltSto = MESASTOC(Close,stoLen)
  smoFiltClose = SUPSMO(Close)
  'buy and exit rules (long only system)
    If CrossesOver(smoFiltSto , lowerVal) And smoFiltClose > smoFiltClose[trndLen]
Then Buy("LE",1,0,Market,Day)
    If CrossesOver(smoFiltSto, upperVal) Then ExitLong("LXob","",1,0,Market,Day)
End Sub

'------------------------------------------------------------
```

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

NINJATRADER: JANUARY 2014 TRADERS' TIPS CODE

The MESA Stochastic, as discussed in "Predictive And Successful Indicators" in this issue by John Ehlers, has been implemented as an indicator available for download at www.ninjatrader.com/SC/January2014SC.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 → Indicator from within the NinjaTrader Control Center window and selecting the “MESA Stochastic” 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 10.

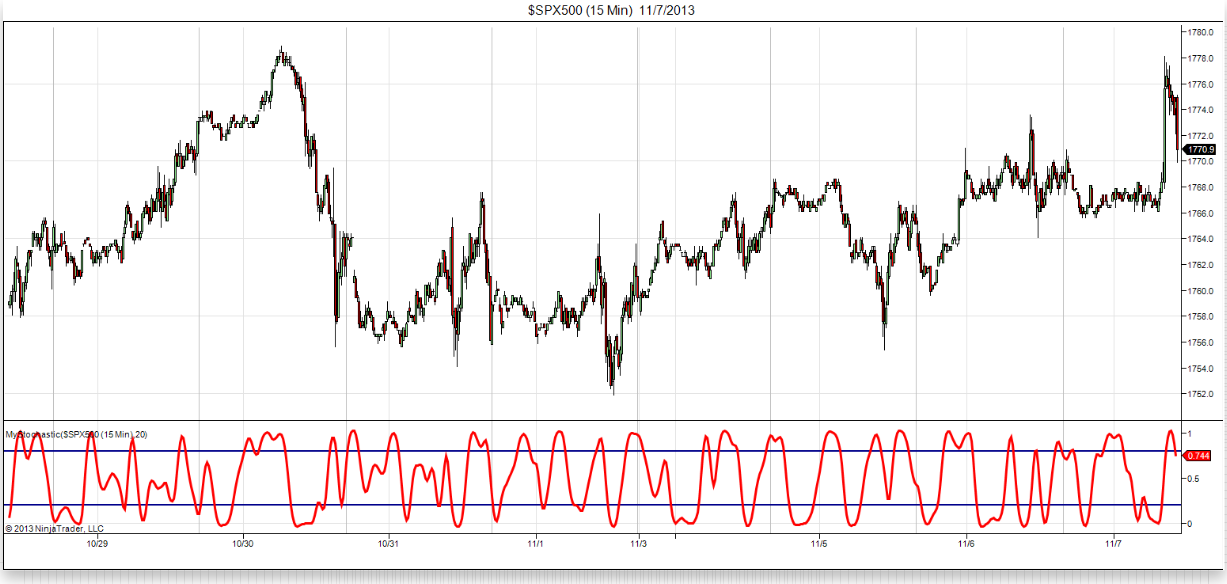


FIGURE 10: NINJATRADER. This screenshot shows the MESA Stochastic applied to a 15-minute chart of the S&P 500 index CFD.

—Raymond Deux & Cal Hueber
NinjaTrader, LLC
www.ninjatrader.com

BACK TO LIST

UPDATA: JANUARY 2014 TRADERS’ TIPS CODE

This tip is based on “Predictive And Successful Indicators” by John Ehlers in this issue. In the article, Ehlers seeks to develop a filter with the optimum smoothing and minimal lag effect, and which mitigates the fractal distortion of the underlying data. A stochastic measure is applied to the resultant series to create an oscillator such that values 0.2 and 0.8 become extremes from which entry signals may be generated. (See Figure 11.)

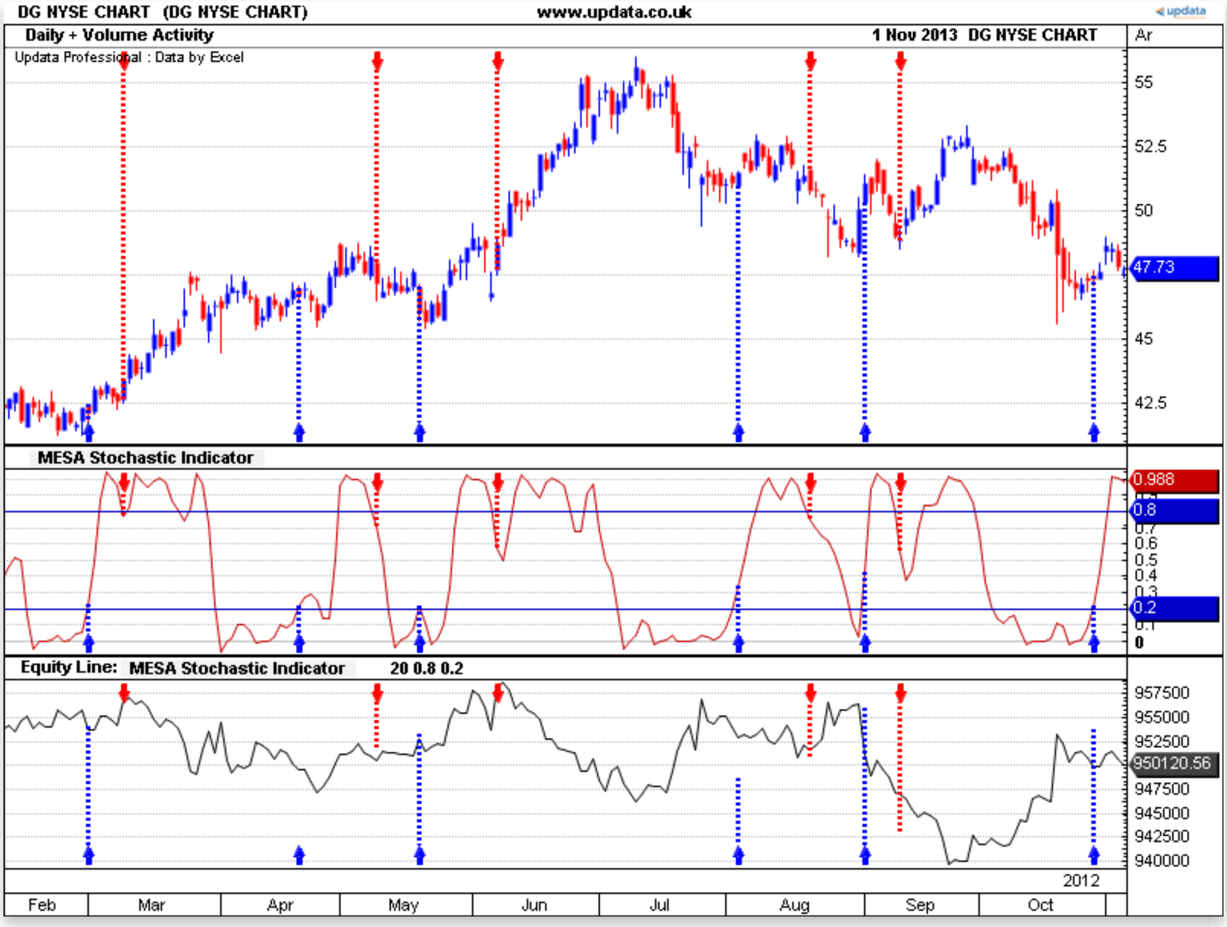


FIGURE 11: UPDATA. This chart shows the stochastic indicator-based system with 0.8/0.2 crossing entries, as applied to Dollar General, in daily resolution data.

The Updata code based on this article is in the Updata Library and may be downloaded by clicking the *custom menu* and *system library*. The code is also shown below and can be pasted into the Updata custom editor and saved.

```
'John F. Ehlers,PhD
'Stocks & Commodities Magazine,January 2014

DISPLAYSTYLE 3LINES
INDICATORTYPE CHART
COLOUR RGB(200,0,0)
COLOUR2 RGB(0,0,200)
COLOUR3 RGB(0,0,200)
PARAMETER "Length" #LENGTH=20
PARAMETER "Upper" @UPPER=0.8
PARAMETER "Lower" @LOWER=0.2
```

```
NAME "MESA Stochastic Indicator" ""
@ALPHA1=0
@HP=0
@A1=0
@B1=0
@C1=0
@C2=0
@C3=0
@FILTER=0
@STOCHASTIC=0
@MESASTOCH=0
FOR #CURDATE=0 TO #LASTDATE
    @ALPHA1=1+(SIN(0.707*2*CONST_PI/48)-1)/COS(0.707*2*CONST_PI/48)
    @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)
    @A1=EXP(-1.414*CONST_PI/10)
    @B1=2*@A1*COS(1.414*CONST_PI/10)
    @C2=@B1
    @C3=-@A1*@A1
    @C1=1-@C2-@C3
    @FILTER=@C1*0.5*(@HP+HIST(@HP,1))+@C2*HIST(@FILTER,1)+@C3*HIST(@FILTER,2)
    @STOCHASTIC=(@FILTER-PLOW(@FILTER,#LENGTH))/(PHIGH(@FILTER,#LENGTH)-
PLOW(@FILTER,#LENGTH))
    @MESASTOCH=@C1*0.5*
(@STOCHASTIC+HIST(@STOCHASTIC,1))+@C2*HIST(@STOCHASTIC,1)+@C3*HIST(@STOCHASTIC,2)
    'OPTIONAL ENTRY RULES
    IF HASX(@MESASTOCH,@UPPER,DOWN)
       SELL CLOSE
       SHORT CLOSE
    ELSEIF HASX(@MESASTOCH,@LOWER,UP)
       COVER CLOSE
       BUY CLOSE
    ENDIF
    @PLOT=@MESASTOCH
    @PLOT2=@UPPER
    @PLOT3=@LOWER
NEXT
```

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

BACK TO LIST

TRADECISION: JANUARY 2014 TRADERS' TIPS CODE

In "Predictive And Successful Indicators" in this issue, author John Ehlers investigates predictive indicators for more effective trading strategies. We are providing code for the three indicators discussed in the article: the SuperSmoother filter, the roofing filter, and the MESA Stochastic indicator.

```
SuperSmoother filter code:
var
a1:=0;
b1:=0;
c1:=0;
c2:=0;
c3:=0;
Filt:=0;
end_var
if HISTORYSIZE < 2 then return 0;

a1:=Exp(-1.414 * 3.14159 / 10);
b1:=2 * a1 * Cos(1.414 * 180 / 10);
c2:=b1;
c3:=-a1 * a1;
c1:=1 - c2 - c3;
Filt:=c1 * (Close + Close\1\) / 2 + c2 * Filt\1\ + c3 * Filt\2\;

return Filt;

Roofing filter code:
var
a1:=0;
b1:=0;
c1:=0;
c2:=0;
c3:=0;
alpha1:=0;
HP:=0;
Filt:=0;
end_var

if HISTORYSIZE < 2 then return 0;

alpha1:=(Cos(0.707 * 360 / 48) + Sin(0.707 * 360 / 48) - 1) / Cos(0.707 * 360 / 48);
HP:=(1 - alpha1 / 2) * (1 - alpha1 / 2) * (Close - 2 * Close\1\ + Close\2\) + 2 * (1
- alpha1) * HP\1\ - (1 - alpha1) * (1 - alpha1) * HP\2\;

a1:=Exp(-1.414 * 3.14159 / 10);
b1:=2 * a1 * Cos(1.414 * 180 / 10);
c2:=b1;
c3:=-a1 * a1;
c1:=1 - c2 - c3;
Filt:=c1 * (HP + HP\1\) / 2 + c2 * Filt\1\ + c3 * Filt\2\;

return Filt;


MESA Stochastic indicator code:
input
Length:"Enter the length:", 20;
end_in
var
a1:=0;
b1:=0;
c1:=0;
```

```
c2:=0;
c3:=0;
alpha1:=0;
HP:=0;
Filt:=0;
HighestC:=0;
LowestC:=0;
count:=0;
Stoc:=0;
MESAStochastic:=0;
end_var

if HISTORYSIZE < 2 then return 0;

alpha1:=(Cos(0.707 * 360 / 48) + Sin(0.707 * 360 / 48) - 1) / Cos(0.707 * 360 / 48);
HP:=(1 - alpha1 / 2) * (1 - alpha1 / 2) * (Close - 2 * Close\1\ + Close\2\) + 2 * (1
- alpha1) * HP\1\ - (1 - alpha1) * (1 - alpha1) * HP\2\;

a1:=Exp(-1.414 * 3.14159 / 10);
b1:=2 * a1 * Cos(1.414 * 180 / 10);
c2:=b1;
c3:=-a1 * a1;
c1:=1 - c2 - c3;
Filt:=c1 * (HP + HP\1\) / 2 + c2 * Filt\1\ + c3 * Filt\2\;

HighestC:=Filt;
LowestC:=Filt;
for count:=0 to Length - 1 do begin
        if Filt\count\ >HighestC then
                HighestC:=Filt\count\;
        if Filt\count\ < LowestC then
                LowestC:=Filt\count\;
end;

Stoc:=(Filt - LowestC) / (HighestC - LowestC);
MESAStochastic:=c1 * (Stoc + Stoc\1\) / 2 + c2 * MESAStochastic\1\ + c3 *
MESAStochastic\2\;
return MESAStochastic;
```

A sample chart plotting the indicators is shown in Figure 12.

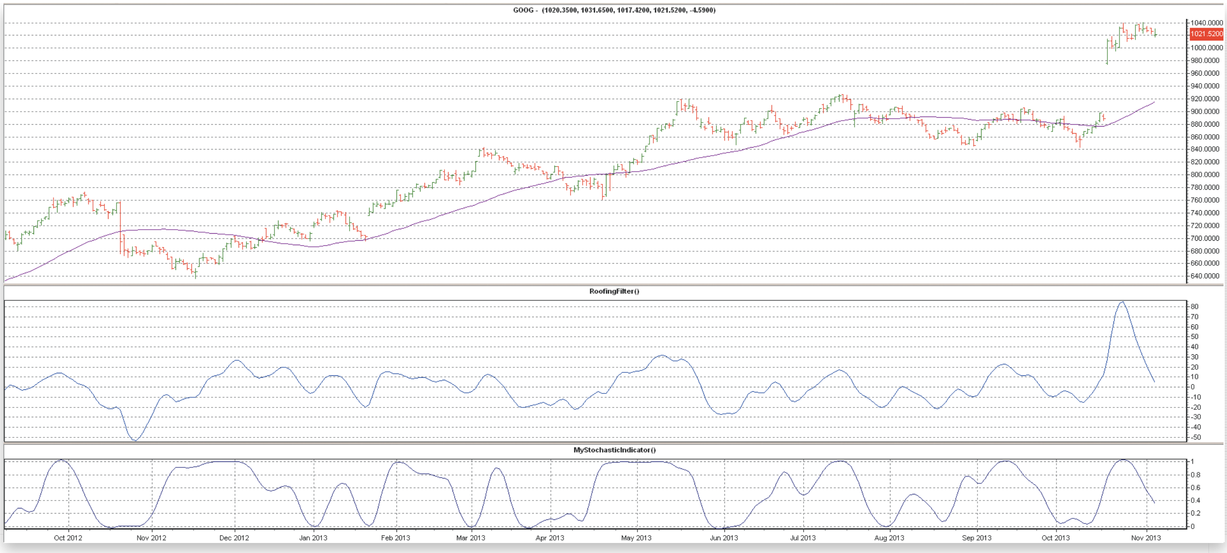


FIGURE 12: TRADECISION. Here is a sample chart of Google with the roofing filter and John Ehlers' MESA Stochastic indicator plotted.

—*Tradecision support*
Tradecision.com

BACK TO LIST

MICROSOFT EXCEL: JANUARY 2014 TRADERS' TIPS CODE

John Ehlers' formulations given in his article in this issue, "Predictive And Successful Indicators," are straightforward to set up in Excel. The chart shown in Figure 13 approximates Ehlers' Figure 8 in his article.

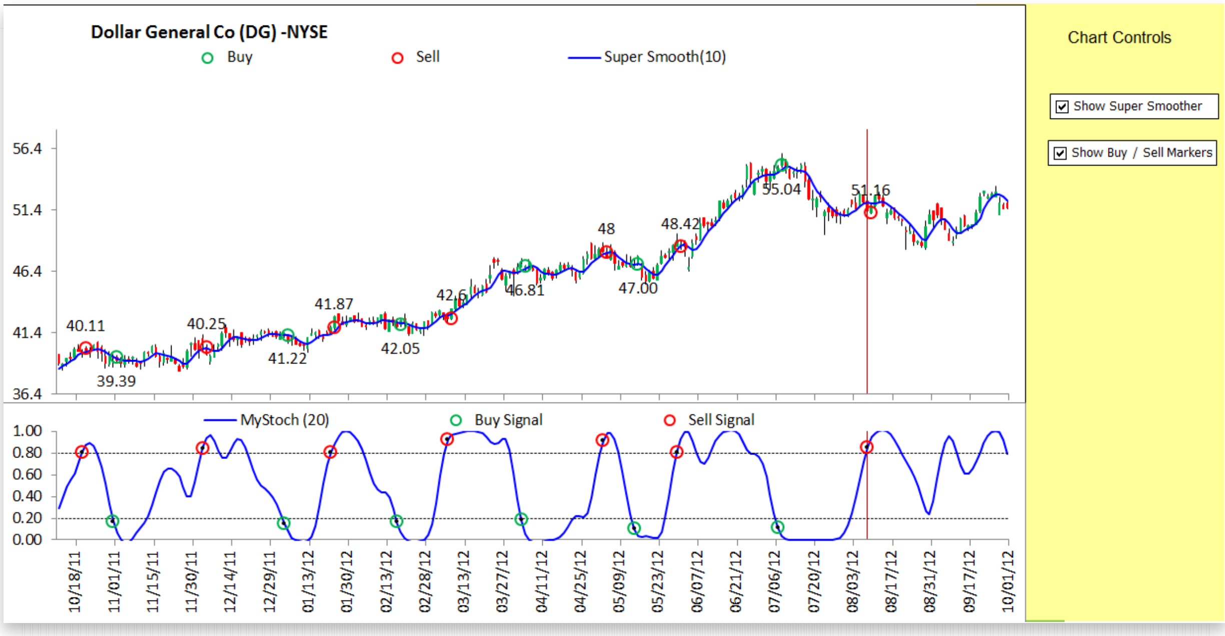
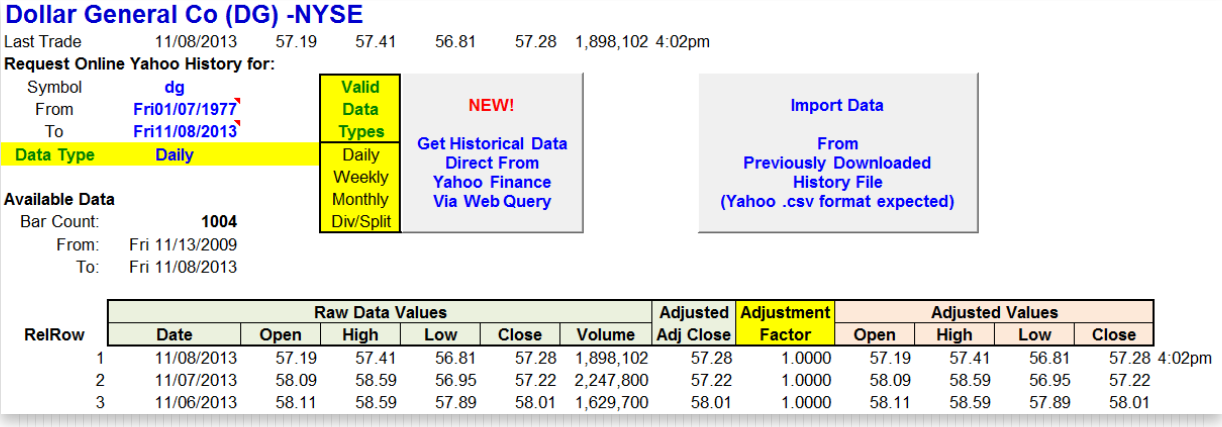


FIGURE 13: EXCEL, SAMPLE SIGNALS. Here is a sample chart of Dollar General with predictive trade signals and one-bar delayed trades.



Dollar General Co (DG) -NYSE

Last Trade 11/08/2013 57.19 57.41 56.81 57.28 1,898,102 4:02pm

Request Online Yahoo History for:

Symbol	dg
From	Fri01/07/1977
To	Fri11/08/2013
Data Type	Daily

Valid Data Types: Daily, Weekly, Monthly, Div/Split

NEW! Get Historical Data Direct From Yahoo Finance Via Web Query

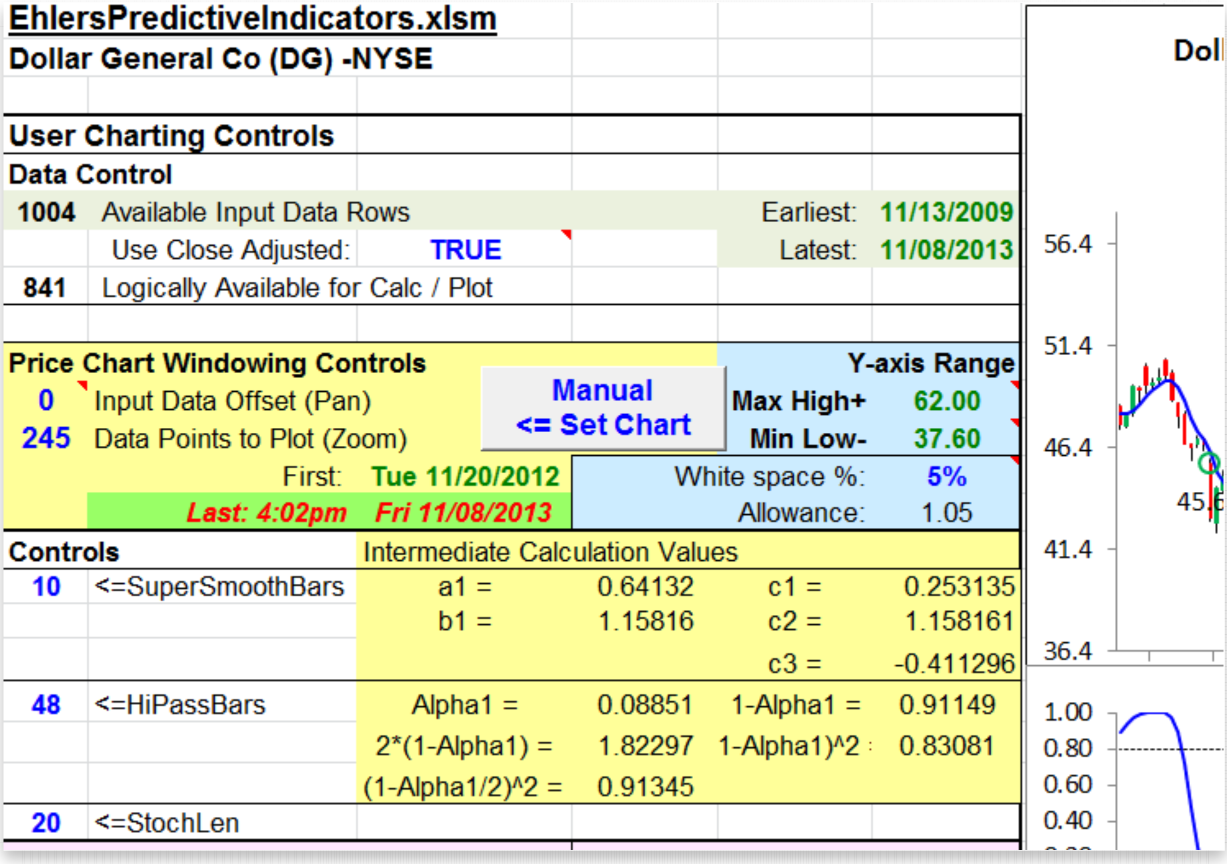
Import Data From Previously Downloaded History File (Yahoo .csv format expected)

Available Data

Bar Count:	1004
From:	Fri 11/13/2009
To:	Fri 11/08/2013

	Raw Data Values						Adjusted	Adjustment	Adjusted Values				
RelRow	Date	Open	High	Low	Close	Volume	Adj Close	Factor	Open	High	Low	Close	
1	11/08/2013	57.19	57.41	56.81	57.28	1,898,102	57.28	1.0000	57.19	57.41	56.81	57.28	4:02pm
2	11/07/2013	58.09	58.59	56.95	57.22	2,247,800	57.22	1.0000	58.09	58.59	56.95	57.22	
3	11/06/2013	58.11	58.59	57.89	58.01	1,629,700	58.01	1.0000	58.11	58.59	57.89	58.01	

FIGURE 14: EXCEL, InputPriceData tab. The InputPriceData tab shows details of the last trade price bar in the second row. This data is inserted into the price history.



EhlersPredictiveIndicators.xlsm

Dollar General Co (DG) -NYSE

User Charting Controls

Data Control

1004 Available Input Data Rows — Earliest: 11/13/2009

Use Close Adjusted: TRUE — Latest: 11/08/2013

841 Logically Available for Calc / Plot

Price Chart Windowing Controls

0 Input Data Offset (Pan)

245 Data Points to Plot (Zoom)

First: Tue 11/20/2012

Last: 4:02pm Fri 11/08/2013

Manual <= Set Chart

Y-axis Range

Max High+ 62.00

Min Low- 37.60

White space %: 5%

Allowance: 1.05

Controls — Intermediate Calculation Values

10 <=SuperSmoothBars

a1 =	0.64132	c1 =	0.253135
b1 =	1.15816	c2 =	1.158161
		c3 =	-0.411296

48 <=HiPassBars

Alpha1 =	0.08851	1-Alpha1 =	0.91149
2*(1-Alpha1) =	1.82297	1-Alpha1)^2 =	0.83081
(1-Alpha1/2)^2 =	0.91345		

20 <=StochLen

FIGURE 15: EXCEL, INTRADAY PRICE BAR TIME STAMP. When the last intraday bar (input data offset = zero) is on the chart, it will be displayed as “Last:”

with a time stamp.

Since this indicator is built on bar closes, transactions cannot logically take place any sooner than the next bar. For trade simulation in this spreadsheet, I enter or exit trades using the open of the next bar after a signal.

This Traders' Tip also introduces a new feature of my Excel templates: an intraday price bar for use with daily price history downloads from Yahoo! Finance.

If bar 1 is an intraday bar, it will have a time stamp to the right.

The intraday bar warning only appears when the bar is on the chart (data window offset = zero).

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" 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 January 2014 issue of
Technical Analysis of STOCKS & COMMODITIES magazine.

TRADERS' TIPS

February 2014

For this month's Traders' Tips, the focus is Oscar Cagigas's article in this issue, "The Degree Of Complexity." Here we present the February 2014 Traders' Tips code with possible implementations in various software.

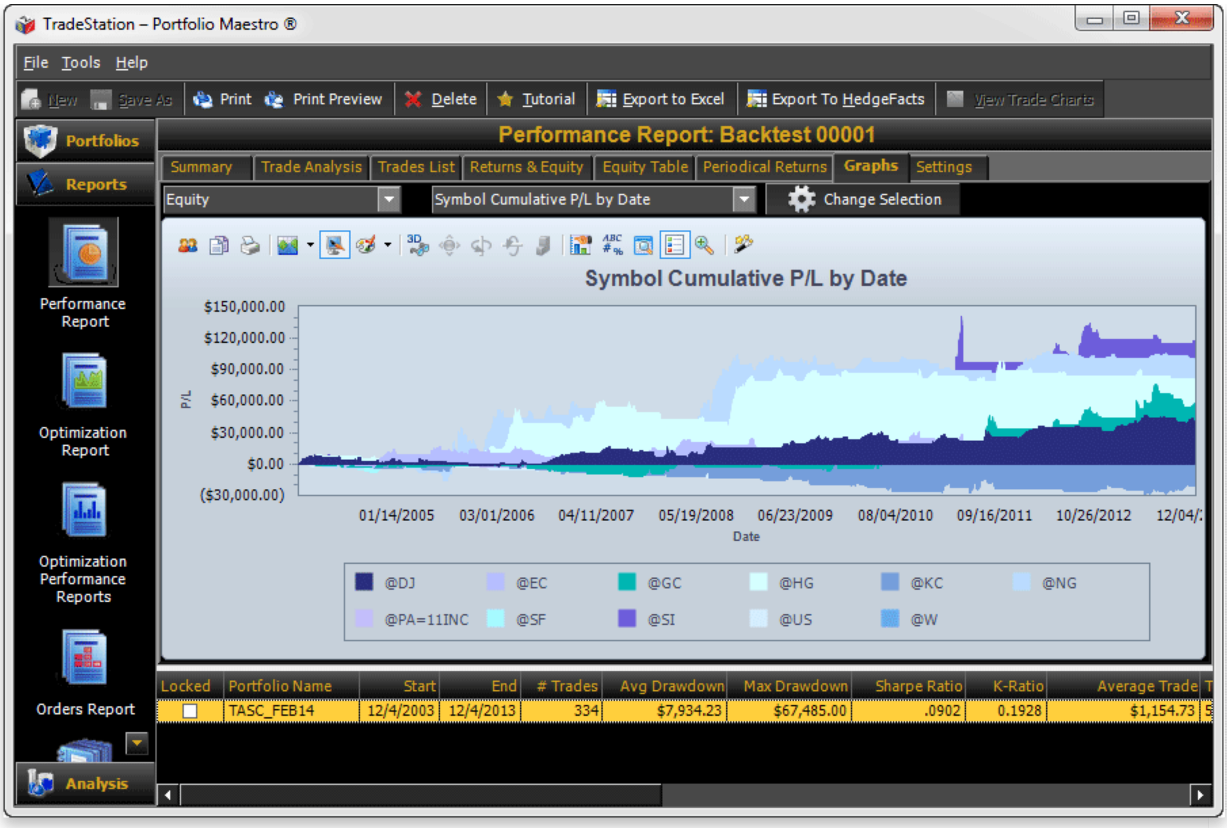
Code for AmiBroker is already provided in Cagigas's article. Subscribers will find that code at the Subscriber Area of our website, www.traders.com. (Click on "Article Code" from the S&C menu.) Presented here is an overview of possible implementations for other software.

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

TRADESTATION: FEBRUARY 2014

In "The Degree Of Complexity" in this issue, author Oscar Cagigas discusses building trading systems and the impact of adding increasing complexity. He presents a simple breakout system based on the well-known Donchian channel. The author tests the strategy performance on a portfolio of 11 commodities over a period of 10 years.

Here, we present some strategy code in EasyLanguage based on Cagigas's basic system (two parameters) as well as the more complex four-parameter system. For this task, we can use TradeStation's Portfolio Maestro (Figure 1) to backtest, optimize, and fully analyze the portfolio.



Locked	Portfolio Name	Start	End	# Trades	Avg Drawdown	Max Drawdown	Sharpe Ratio	K-Ratio	Average Trade	T
☐	TASC_FEB14	12/4/2003	12/4/2013	334	$7,934.23	$67,485.00	.0902	0.1928	$1,154.73	5

FIGURE 1: TRADESTATION, CUMULATIVE P/L. Using TradeStation's Portfolio Maestro to backtest, optimize, and analyze a sample portfolio, here is a sample cumulative profit & loss (P/L) over the period 2003–2013 by security.

To download the EasyLanguage code, please visit our TradeStation and EasyLanguage support forum. The EasyLanguage code based on Cagigas's article can be found here: http://www.tradestation.com/TASC-2014. The ELD filename is "_TASC_DonchianSystem.ELD."

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

```
_TASC_DonchianSystem (Indicator)

{Reference: Technical Analysis of Stocks & Commodities,
Feb 2014.
Article: The Degree Of Complexity

Basic and Complex 4 parameter system

Usage Notes:
To test the 'Basic' System set the following
inputs to:

ATRVolCoef = 0
ATRStopLossMult = 0
}
```

```
inputs:
        int EntryChannelLength( 40 ),
        int ExitChannelLength( 15 ),
        int ATRLength( 20 ),
        double ATRVolCoef( .9 ),
        double ATRStopLossMult( 4 ) ;

variables:
        double UpperEntryChannel( 0 ),
        double LowerEntryChannel( 0 ),
        double UpperExitChannel( 0 ),
        double LowerExitChannel( 0 ),
        double StopATR( 0 ),
        bool EntryVolOK( false ) ;


UpperEntryChannel = Highest( High, EntryChannelLength ) ;
LowerEntryChannel = Lowest( Low, EntryChannelLength ) ;
UpperExitChannel = Highest( High, ExitChannelLength ) ;
LowerExitChannel = Lowest( Low, ExitChannelLength ) ;

StopATR = AvgTrueRange( ATRLength ) ;
EntryVolOK = AvgTrueRange( 1 ) < StopATR[1] * ATRVolCoef ;

if EntryVolOK or ATRVolCoef = 0 then
        begin
        Buy next bar at UpperEntryChannel Stop ;
        SellShort next bar at LowerEntryChannel Stop ;
        end ;

if MarketPosition = 1 then
        begin
        Sell next bar at LowerExitChannel Stop ;
        if ATRStopLossMult <> 0 then
                Sell next bar at
                EntryPrice - StopATR * ATRStopLossMult Stop ;
        end
else if MarketPosition = -1 then
        begin
        BuyToCover next bar at UpperExitChannel Stop ;
        if ATRStopLossMult <> 0 then
                BuyToCover next bar at
                EntryPrice + StopATR * ATRStopLossMult Stop ;
        end ;
```

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.

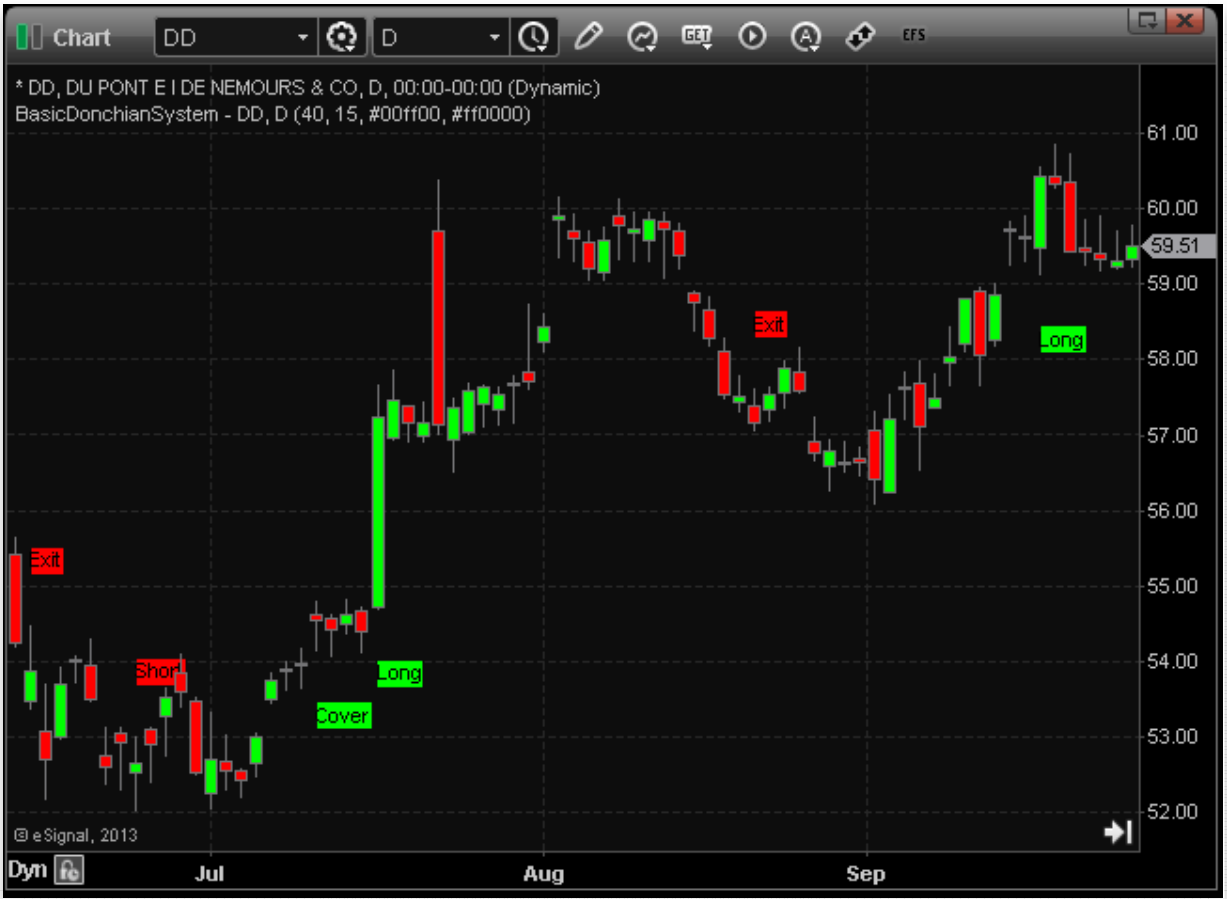
—Doug McCrary
TradeStation Securities, Inc.
www.TradeStation.com

BACK TO LIST

eSIGNAL: FEBRUARY 2014

For this month's Traders' Tip, we've provided the formula BasicDonchianSystem.efs based on the formulas described in Oscar Cagigas's article in this issue, "The Degree Of Complexity."

The study contains formula parameters for *Len1*, *Len2*, long color, and short color, all of which may be configured through the *edit chart* window (right-click on "chart" and select *edit chart*). A sample chart is shown in Figure 2.



* DD, DU PONT E I DE NEMOURS & CO, D, 00:00-00:00 (Dynamic)
BasicDonchianSystem - DD, D (40, 15, #00ff00, #ff0000)

FIGURE 2: eSIGNAL SAMPLE CHART

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 ?© 2013)
    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 Degree Of Complexity by Oscar G. Cagigas

Formula Parameters:                     Default:
Len1                                    40

Len2                                    15

Long Color                              lime

Short Color                             red


Version:            1.00  12/05/2013

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("BasicDonchianSystem");

    setPriceStudy(true);

    var x = 0;

    fpArray[x] = new FunctionParameter("fpLen1", FunctionParameter.NUMBER);
    with(fpArray[x++])
    {
        setName("Len1");
        setLowerLimit(20);

        setUpperLimit(40);
        setDefault(40);
    }



    fpArray[x] = new FunctionParameter("fpLen2", FunctionParameter.NUMBER);
    with(fpArray[x++])
    {
        setName("Len2");
        setLowerLimit(5);

        setUpperLimit(30);
        setDefault(15);
    }
    fpArray[x] = new FunctionParameter("gLongColor", FunctionParameter.COLOR);
    with(fpArray[x++])
    {
        setName("Long Color");
        setDefault(Color.lime);
    }

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

var bInit = false;
var bVersion = null;

var xHHV1 = null;

var xLLV1 = null;



var xHHV2 = null;

var xLLV2 = null;



var xHigh = null;

var xLow  = null;

var xOpen = null;


function main(fpLen1,fpLen2,gLongColor,gShortColor)
{
    if (bVersion == null) bVersion = verify();
    if (bVersion == false) return;

    if (!bInit)
    {

        xHigh = high();

        xLow = low();

        xOpen = open();



        xHHV1 = hhv(fpLen1, xHigh);

        xLLV1 = llv(fpLen1, xLow);



        xHHV2 = hhv(fpLen2, xHigh);
        xLLV2 = llv(fpLen2, xLow);

        bInit = true;
    }

    var nBuyStop   = xHHV1.getValue(-1);

    var nSellStop  = xLLV2.getValue(-1);

    var nShortStop = xLLV1.getValue(-1);

    var nCoverStop = xHHV2.getValue(-1);


    var nHigh = xHigh.getValue(0);

    var nLow  = xLow.getValue(0);

    var nOpen = xOpen.getValue(0);


    if (nBuyStop == null || nSellStop == null || nShortStop == null || nCoverStop ==
null || nHigh == null || nLow == null || nOpen == null)

        return;


    nPrice=null;


    if (getCurrentBarIndex() != 0)
    {

            // Entry Strategy
            if (!Strategy.isInTrade())
            {
               //LONG ENTRY

                    if (nHigh > nBuyStop)

                    {

                       Strategy.doLong("Enter Long", Strategy.CLOSE,
Strategy.THISBAR);


                       drawTextRelative(0, BelowBar2, "Long", Color.black,
gLongColor, Text.PRESET, null, null);
                    }

                else

                {
                    //SHORT ENTRY

                    if (nLow < nShortStop)
                    {

                       Strategy.doShort("Enter Short", Strategy.CLOSE,
Strategy.THISBAR);


                       drawTextRelative(0, AboveBar2, "Short", Color.black,
gShortColor, Text.PRESET, null, null);
                    }

                }
            }



            // Exit Strategy
            if (Strategy.isInTrade())
            {

                //EXIT

                if ((nLow < nSellStop) && Strategy.isLong())
                        {

                           Strategy.doSell("Exit", Strategy.CLOSE,
Strategy.THISBAR);


                           drawTextRelative(0, AboveBar2, "Exit", Color.black,
gShortColor, Text.PRESET, null, null);
                        }

                else

                    {

                        //COVER

                        if ((nHigh > nCoverStop) && Strategy.isShort())
                        {

                           Strategy.doCover("Cover", Strategy.CLOSE,
Strategy.THISBAR);


                           drawTextRelative(0, BelowBar2, "Cover", Color.black,
gLongColor, Text.PRESET, null, null);
                        }

                }

            }
        }


    return;
```

```
}

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;
}
```

—Jason Keck
eSignal, an Interactive Data company
800 779-6555, www.eSignal.com

BACK TO LIST

THINKORSWIM: FEBRUARY 2014

In "The Degree Of Complexity" in this issue, author Oscar Cagigas describes how to build a trading system with a low level of complexity. For thinkorswim users, we have recreated this strategy in our proprietary scripting language, thinkScript. To get the strategy, simply go to: http://tos.mx/s7sg5S

You can adjust the parameters within the *edit studies* window to fine-tune your variables. A sample chart implementation is shown in Figure 3.

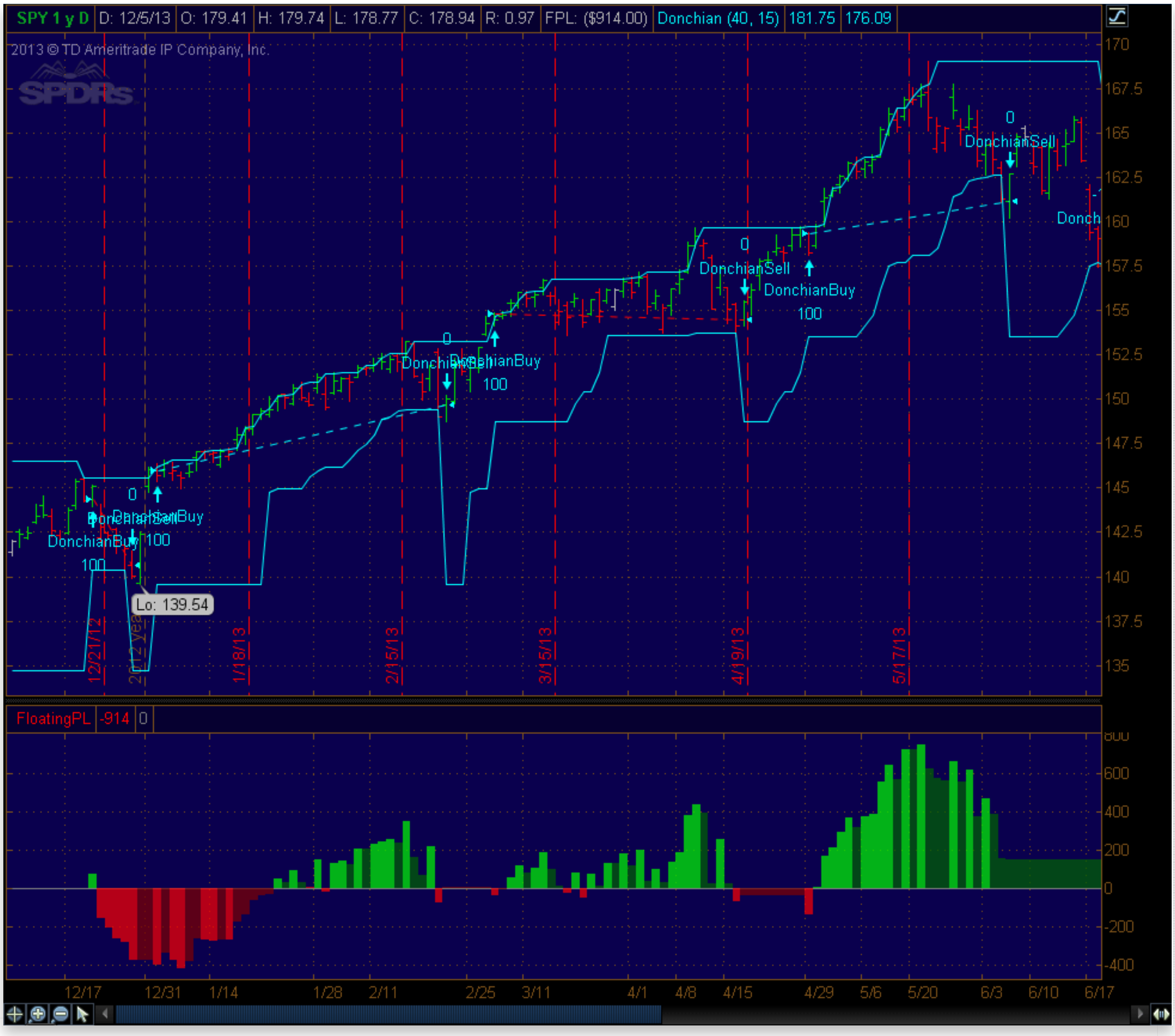


FIGURE 3: THINKORSWIM

Alternatively, you can manually load the strategy code by following these steps:

1. From TOS charts, select Studies → Edit studies
2. Select the *strategy* tab in the upper left-hand corner
3. Select "new" in the lower left-hand corner
4. Name the strategy (for example, "DonchianStrategy")
5. Click in the script editor window, remove "addOrder(OrderType.BUY_AUTO, no);" and paste in the following code:

```
input length1 = 40;
input length2 = 15;

def buystop = Highest(high, length1)[1];
def sellstop = Lowest(low, length2)[1];
def shortstop = Lowest(low, length1)[1];
def coverstop = Highest(high, length2)[1];

def buy = high > buystop;
def short = low < shortstop;
def sell = low < sellstop;
def cover = high > coverstop;

def position = {default none, long, short};
position = if (buy or (position[1] == position.long and !sell)) then
position.long else if (short or (position[1] == position.short and !cover)) then
position.short else position.none;

plot BandUp = if position[1] == position.short then coverstop else buystop;
plot BandDown = if position[1] == position.long then sellstop else shortstop;
BandUp.SetDefaultColor(GetColor(1));
BandDown.SetDefaultColor(GetColor(1));

AddOrder(OrderType.BUY_AUTO, buy, tickColor = GetColor(1), arrowColor =
GetColor(1), name = "DonchianBuy");
AddOrder(OrderType.SELL_AUTO, short, tickColor = GetColor(1), arrowColor =
GetColor(1), name = "DonchianShort");
AddOrder(OrderType.SELL_TO_CLOSE, sell, tickColor = GetColor(1), arrowColor =
GetColor(1), name = "DonchianSell");
AddOrder(OrderType.BUY_TO_CLOSE, cover, tickColor = GetColor(1), arrowColor =
GetColor(1), name = "DonchianCover");
```

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

BACK TO LIST

MetaStock

METASTOCK: FEBRUARY 2014

Oscar Cagigas's article in this issue, "The Degree Of Complexity," discusses how simpler systems can perform better than more complex ones. Formulas for MetaStock to implement Cagigas's basic Donchian channel system are provided here.

```
Buy Order:

Len1:= OPT1;
Len2:= OPT2;
el:= h > ref(hhv(H,len1),-1);
es:= L < ref(llv(L,len1),-1);
xl:= L < ref(llv(L,len2),-1);
xs:= h > ref(hhv(H,len2),-1);
trade:= if(el, 1, if(es, -1,
if((prev=1 and xl) or (prev=-1 and xs), 0, prev)));
trade=1 and ref(trade<>1, -1)

Sell Order:

Len1:= OPT1;
```

```
Len2:= OPT2;
el:= h > ref(hhv(H,len1),-1);
es:= L < ref(llv(L,len1),-1);
xl:= L < ref(llv(L,len2),-1);
xs:= h > ref(hhv(H,len2),-1);
trade:= if(el, 1, if(es, -1,
if((prev=1 and xl) or (prev=-1 and xs), 0, prev)));
trade=0 and ref(trade=1, -1)
```

Sell Short Order:

```
Len1:= OPT1;
Len2:= OPT2;
el:= h > ref(hhv(H,len1),-1);
es:= L < ref(llv(L,len1),-1);
xl:= L < ref(llv(L,len2),-1);
xs:= h > ref(hhv(H,len2),-1);
trade:= if(el, 1, if(es, -1,
if((prev=1 and xl) or (prev=-1 and xs), 0, prev)));
trade=-1 and ref(trade<>-1, -1)
```

Buy to Cover Order:

```
Len1:= OPT1;
Len2:= OPT2;
el:= h > ref(hhv(H,len1),-1);
es:= L < ref(llv(L,len1),-1);
xl:= L < ref(llv(L,len2),-1);
xs:= h > ref(hhv(H,len2),-1);
trade:= if(el, 1, if(es, -1,
if((prev=1 and xl) or (prev=-1 and xs), 0, prev)));
trade=0 and ref(trade=-1, -1)
```

Optimizations:

```
OPT1:
Description:  long term
Minimum: 20
Maximum: 40
Step : 5

OPT2:
Description:  short term
Minimum: 5
Maximum: 20
Step : 5
```

—William Golson
MetaStock Technical Support
www.metastock.com

BACK TO LIST

NEUROSHELL TRADER: FEBRUARY 2014

The Donchian channel trading system described by Oscar Cagigas in his article in this issue, “The Degree Of Complexity,” 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 NeuroShell trading strategy wizard:

Generate a buy long STOP order if all of the following are true:

```
High Channel Breakout(High,40)
Stop Price: Lag(Max(High,40),1)
```

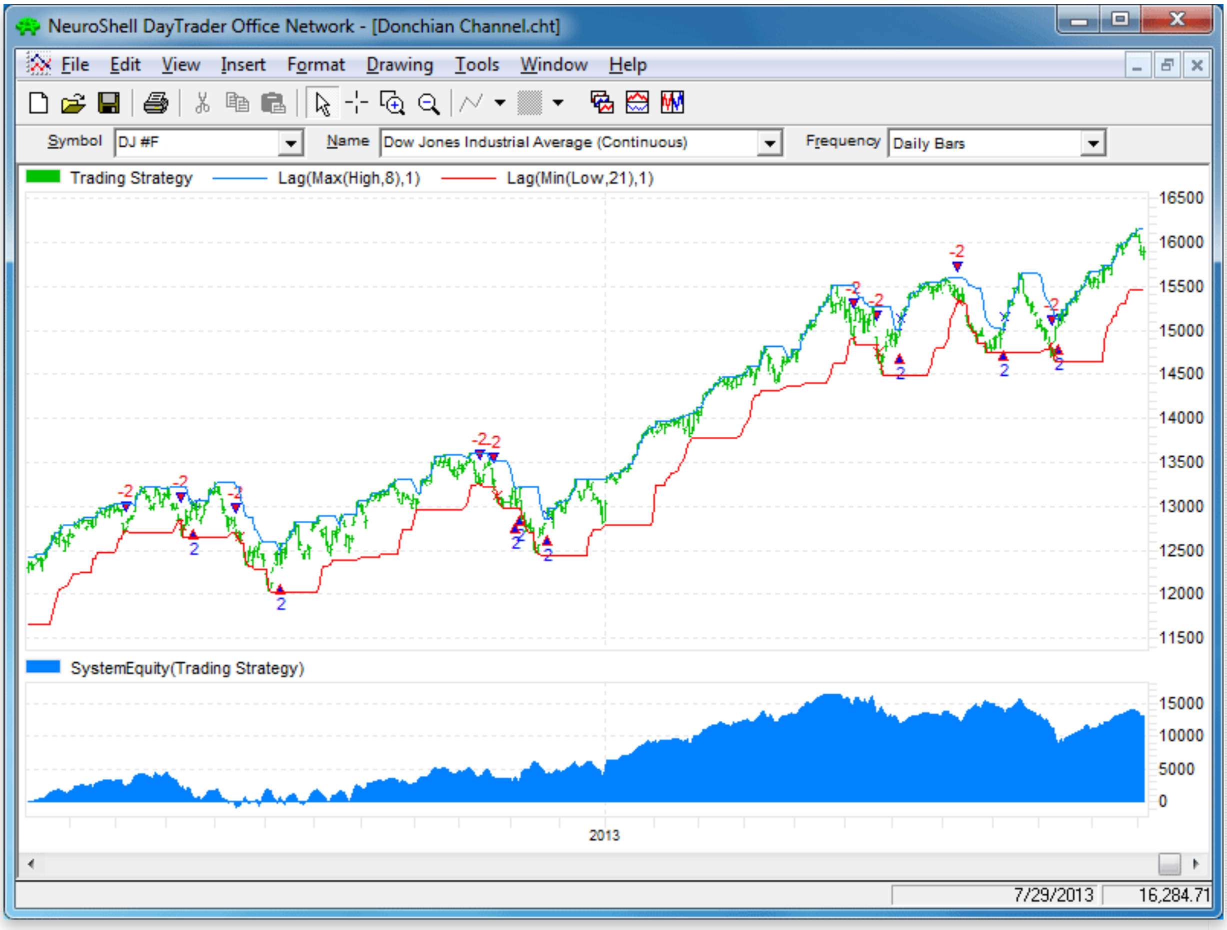
Generate a sell short STOP order if all of the following are true:

```
Low Channel Breakout(Low,15)
Stop Price: Lag(Min(Low,15),1)
```

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 website to download a copy of this or any previous Traders’ Tips.

A sample chart is shown in Figure 4.



Symbol DJ #F

Name Dow Jones Industrial Average (Continuous)

Frequency Daily Bars

FIGURE 4: NEUROSHELL TRADER. This NeuroShell Trader chart displays the out-of-sample Donchian channel trading after optimizing on prior data.

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

BACK TO LIST

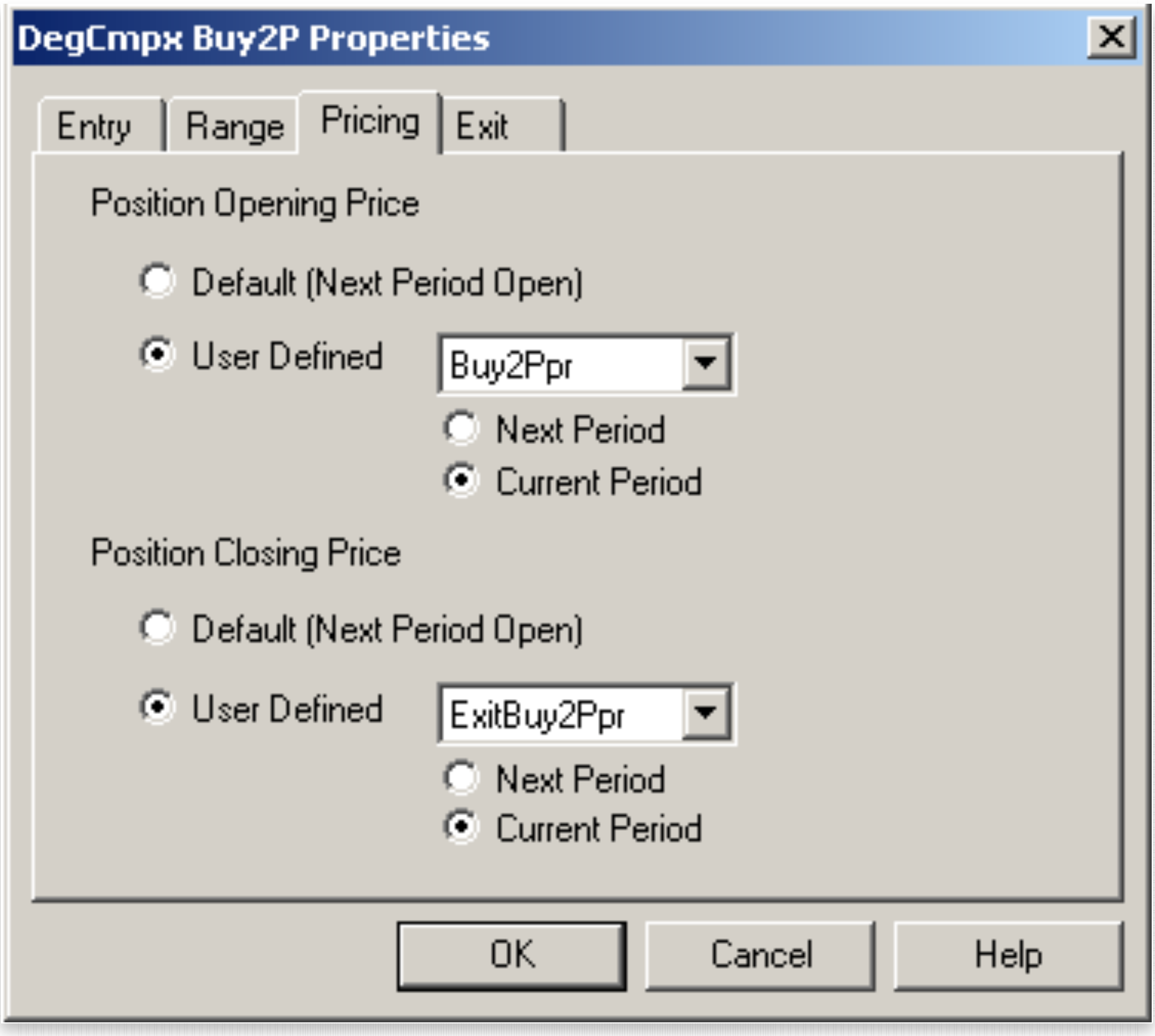
AIQ: FEBRUARY 2014

This Traders' Tip is based on "The Degree Of Complexity" by Oscar Cagigas in this issue.

The five Donchian breakout systems described by Cagigas in his article are coded using the following rules:

System	Entry rule	Exit rule	Entry price formula	Exit price formula
2 Param-Long	Buy2P	ExitBuy2P	Buy2Ppr	ExitBuy2Ppr
2 Param-Short	Sell2P	ExitSell2P	Sell2Ppr	ExitSell2Ppr
4 Param-Long	Buy4P	ExitBuy4P	Buy4Ppr	ExitBuy4Ppr
4 Param-Short	Sell4P	ExitSell4P	ell4Ppr	ExitSell4Ppr
6 Param-Long	Buy6P	ExitBuy6P	Buy6Ppr	ExitBuy6Ppr
6 Param-Short	Sell6P	ExitSell6P	Sell6Ppr	ExitSell6Ppr
8 Param-Long	Buy8P	ExitBuy8P	Buy8Ppr	ExitBuy8Ppr
8 Param-Short	Sell8P	ExitSell8P	Sell8Ppr	ExitSell8Ppr
9 Param-Long	Buy9P	ExitBuy9P	Buy9Ppr	ExitBuy9Ppr
9 Param-Short	Sell9P	ExitSell9P	Sell9Ppr	ExitSell9Ppr

The EDS code file has the backtests already set up for all of these long and short rules. In Figure 5, I show a typical setup for the pricing. In Figure 6, I show a typical setup for the exit rule.



Position Opening Price

○ Default (Next Period Open)

● User Defined Buy2Ppr

○ Next Period

● Current Period

Position Closing Price

○ Default (Next Period Open)

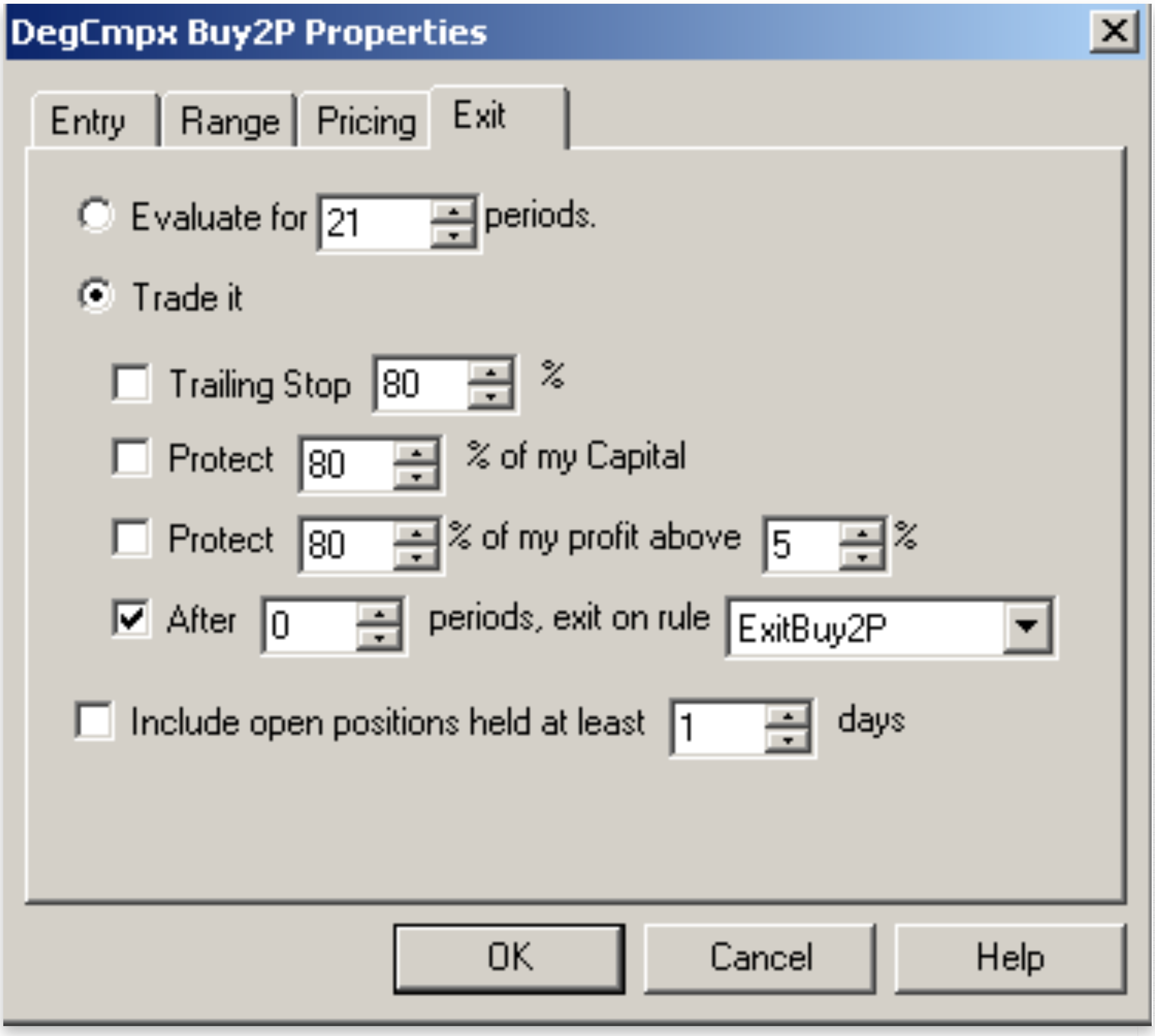
● User Defined ExitBuy2Ppr

○ Next Period

● Current Period

FIGURE 5: AIQ. Here is a typical setup for the pricing

portion of the backtests.



○ Evaluate for 21 periods.

● Trade it

□ Trailing Stop 80 %

□ Protect 80 % of my Capital

□ Protect 80 % of my profit above 5 %

☑ After 0 periods, exit on rule ExitBuy2P

□ Include open positions held at least 1 days

FIGURE 6: AIQ. Here is a typical setup for the exit portion of the backtests.

The AIQ code and EDS file can be downloaded from www.TradersEdgeSystems.com/traderstips.htm, and is also shown below.

```
!THE DEGREE OF COMPLEXITY

!Author: Oscar G. Cagigas, TASC Feb 2014

!Coded by Richard Denning 12/7/13

!www.TradersEdgeSystems.com


! CODING ABBREVIATIONS:
```

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


! PARAMETERS:
donLen1 is 40.
donLen2 is 15.
atrLen  is 20.
atrMult is 1.0.
atrStop         is 4.0.
maLen1  is 10.
maLen2  is 100.
rsiLen  is 14.
rsiUpper        is 70.
rsiLower        is 30.
atrMult2        is 0.6.
minMov  is 0.01.
!------------------------------------------------------------------
! AVERAGE TRUE RANGE (AS DEFINED BY WELLS WILDER)
WWE     is 2*(atrLen-1).
TR      is Max(H - L,max(abs(valresult(C,1) - L),abs(valresult(C,1)- H))).
ATR     is expAvg(TR,WWE).
ATR1    is valresult(ATR,1).
!------------------------------------------------------------------
!! RSI WILDER
!To convert Wilder Averaging to Exponential Averaging use this formula:
!ExponentialPeriods = 2 * WilderPeriod - 1.
```

```
U       is C-valresult(C,1).

D       is valresult(C,1)-C.

eLen    is 2 * rsiLen - 1.

AvgU    is ExpAvg(iff(U>0,U,0),eLen).

AvgD    is ExpAvg(iff(D>=0,D,0),eLen).

rsi     is 100-(100/(1+(AvgU/AvgD))).

!-----------------------------------------------------------------

!BASIC SYSTEM 2P (2 parameters):

HHdL1 is  highresult(H,donLen1,1).

HHdL2 is  highresult(H,donLen2,1).

LLdL1 is lowresult(L,donLen1,1).

LLdL2 is lowresult(L,donLen2,1).

Buy2P if H > HHdL1.

ExitBuy2P if  L < LLdL2.

Sell2P if L < LLdL1.

ExitSell2P if H > HHdL2.

Buy2Ppr is max(O,HHdL1 + minMov).

ExitBuy2Ppr is iff(ExitBuy2P,min(O,LLdL2 - minMov),C).

Sell2Ppr is min(O,LLdL1 - minMov).

ExitSell2Ppr is max(O,HHdL2 + minMov).

!------------------------------------------------------------------

!FOUR PARAMETER SYSTEM 4P:

Buy4P if H > HHdL1 and valrule(TR < atrMult*ATR,1).

EB4P1 if L < LLdL2.

EB4Pval is PEP - atrStop*valresult(ATR,PD).

EB4P2 if L < valresult(EB4Pval,1).

ExitBuy4P if EB4P1 or EB4P2.

Sell4P if L < LLdL1 and valrule(TR < atrMult*ATR,1).

ES4Pval is PEP + atrStop*valresult(ATR,PD) .

ES4P1 if H > HHdL2.
```

```
ES4P2 if H > valresult(ES4Pval,1).
ExitSell4P if ES4P1 or ES4P2.
Buy4Ppr is max(O,HHdL1 + minMov).
ExitBuy4Ppr is iff(EB4P1,min(O,LLdL2 - minMov),
        iff(EB4P2,min(O,valresult(EB4Pval - minMov,1)),-99)).
Sell4Ppr is min(O,LLdL1 - minMov).
ExitSell4Ppr is iff(ES4P1,max(O,HHdL2 + minMov),
        iff(ES4P2,max(O,valresult(ES4Pval + minMov,1)),-99)).
!-----------------------------------------------------------------
!SIX PARAMETER SYSTEM 6P:
SMA1 is simpleavg(C,maLen1).
SMA2 is simpleavg(C,maLen2).
Buy6P if H > HHdL1 and valrule(TR < atrMult*ATR,1)
        and valrule(SMA1 > SMA2,1).
EB6P1 if L < LLdL2.
EB6Pval is PEP - atrStop*valresult(ATR,PD).
EB6P2 if L < valresult(EB6Pval,1).
ExitBuy6P if EB6P1 or EB6P2.
Sell6P if L < LLdL1 and valrule(TR < atrMult*ATR,1)
        and valrule(SMA1 < SMA2,1).
ES6Pval is PEP + atrStop*valresult(ATR,PD) .
ES6P1 if H > HHdL2.
ES6P2 if H > valresult(ES6Pval,1).
ExitSell6P if ES6P1 or ES6P2.
Buy6Ppr is max(O,HHdL1 + minMov).
ExitBuy6Ppr is iff(EB6P1,min(O,LLdL2 - minMov),
        iff(EB6P2,min(O,valresult(EB6Pval - minMov,1)),-99)).
Sell6Ppr is min(O,LLdL1 - minMov).
ExitSell6Ppr is iff(ES6P1,max(O,HHdL2 + minMov),
        iff(ES6P2,max(O,valresult(ES6Pval + minMov,1)),-99)).
```

```
!---------------------------------------------------------------

!EIGHT PARAMETER SYSTEM 8P:

Buy8P if H > HHdL1 and valrule(TR < atrMult*ATR,1)

        and valrule(SMA1 > SMA2,1) and valrule(RSI >= rsiUpper,1).

EB8P1 if L < LLdL2.

EB8Pval is PEP - atrStop*valresult(ATR,PD).

EB8P2 if L < valresult(EB8Pval,1).

ExitBuy8P if EB8P1 or EB8P2.

Sell8P if L < LLdL1 and valrule(TR < atrMult*ATR,1)

        and valrule(SMA1 < SMA2,1) and valrule(RSI <= rsiLower,1).

ES8Pval is PEP + atrStop*valresult(ATR,PD) .

ES8P1 if H > HHdL2.

ES8P2 if H > valresult(ES8Pval,1).

ExitSell8P if ES8P1 or ES8P2.

Buy8Ppr is max(O,HHdL1 + minMov).

ExitBuy8Ppr is iff(EB8P1,min(O,LLdL2 - minMov),

        iff(EB8P2,min(O,valresult(EB8Pval - minMov,1)),-99)).

Sell8Ppr is min(O,LLdL1 - minMov).

ExitSell8Ppr is iff(ES8P1,max(O,HHdL2 + minMov),

        iff(ES8P2,max(O,valresult(ES8Pval + minMov,1)),-99)).

!----------------------------------------------------------------

!NINE PARAMETER SYSTEM 9P:

Buy9P if H > HHdL1 and valrule(TR < atrMult*ATR,1)

        and valrule(SMA1 > SMA2,1) and valrule(RSI >= rsiUpper,1)

        and valrule(TR > atrmult2*ATR,1) .

EB9P1 if L < LLdL2.

EB9Pval is PEP - atrStop*valresult(ATR,PD).

EB9P2 if L < valresult(EB9Pval,1).

ExitBuy9P if EB9P1 or EB9P2.

Sell9P if L < LLdL1 and valrule(TR < atrMult*ATR,1)
```

```
        and valrule(SMA1 < SMA2,1) and valrule(RSI <= rsiLower,1)
        and valrule(TR > atrmult2*ATR,1) .
ES9Pval is PEP + atrStop*valresult(ATR,PD) .
ES9P1 if H > HHdL2.
ES9P2 if H > valresult(ES9Pval,1).
ExitSell9P if ES9P1 or ES9P2.
Buy9Ppr is max(O,HHdL1 + minMov).
ExitBuy9Ppr is iff(EB9P1,min(O,LLdL2 - minMov),
        iff(EB9P2,min(O,valresult(EB9Pval - minMov,1)),-99)).
Sell9Ppr is min(O,LLdL1 - minMov).
ExitSell9Ppr is iff(ES9P1,max(O,HHdL2 + minMov),
        iff(ES9P2,max(O,valresult(ES9Pval + minMov,1)),-99)).
!----------------------------------------------------------------
```

—Richard Denning
info@TradersEdgeSystems.com
for AIQ Systems

BACK TO LIST

TRADERSSTUDIO: FEBRUARY 2014

The TradersStudio code for Oscar Cagigas's systems presented in his article in this issue, "The Degree Of Complexity," can be found at:

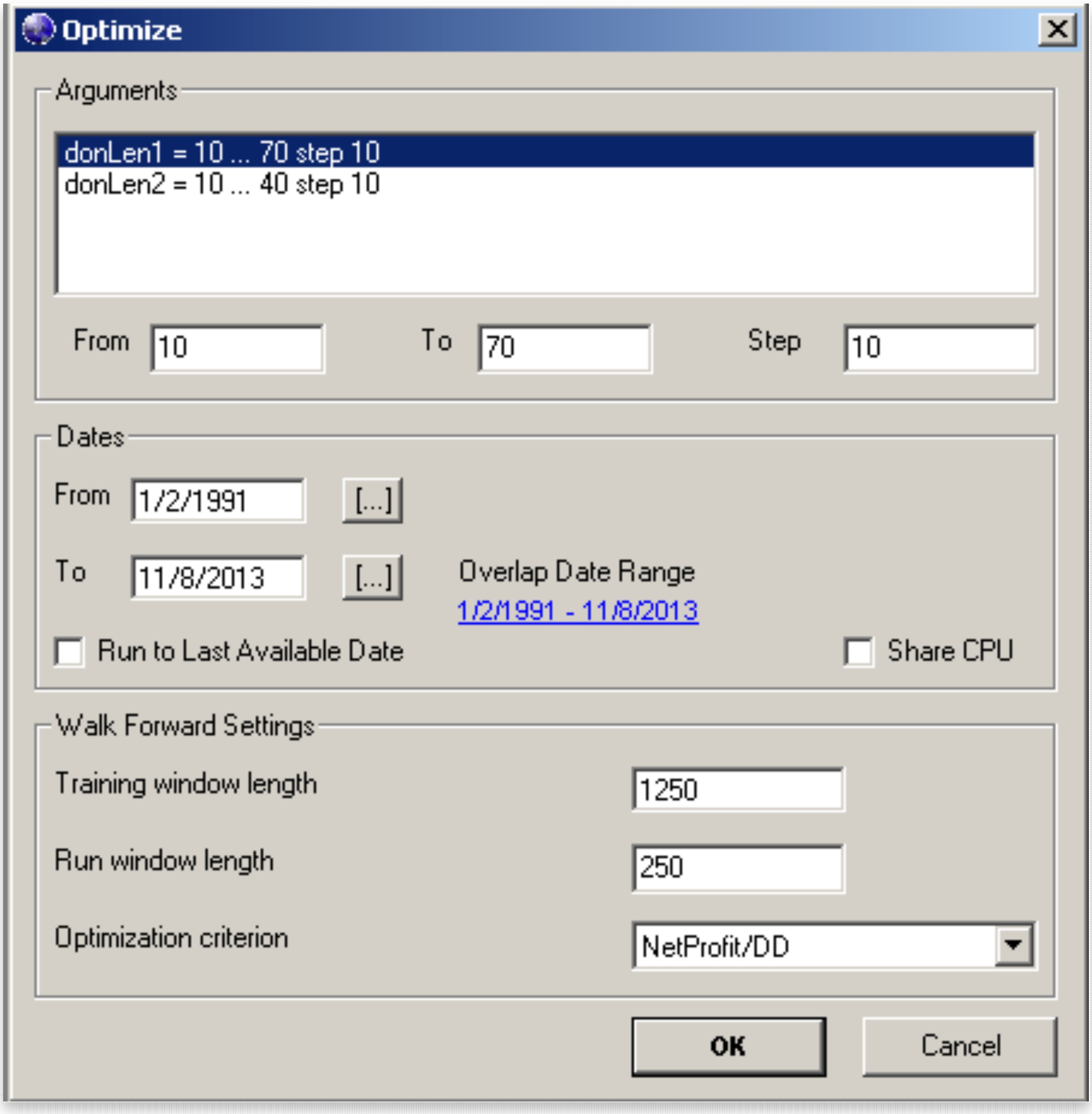
- www.TradersEdgeSystems.com/traderstips.htm
- www.TradersStudio.com → Traders Resources → Traders' Tips

The following code files are contained in the download:

- System: DON2P – Basic Donchian system with two parameters
- System: DON4P – Donchian system with four parameters
- System: DON6P – Donchian system with six parameters
- System: DON8P – Donchian system with eight parameters

- System: DON9P – Donchian system with nine parameters

The in-sample and out-of-sample tests discussed by Cagigas could be run using TradersStudio's built-in walk-forward processor. Figure 7 shows the typical setup in TradersStudio to run a walk-forward optimization for the DON2P system.


Optimize

Arguments

donLen1 = 10 ... 70 step 10
donLen2 = 10 ... 40 step 10

From 10 To 70 Step 10

Dates

From 1/2/1991 [...]

To 11/8/2013 [...]

Overlap Date Range
1/2/1991 - 11/8/2013

□ Run to Last Available Date

□ Share CPU

Walk Forward Settings

Training window length 1250

Run window length 250

Optimization criterion NetProfit/DD

FIGURE 7: TRADERSSTUDIO. Here is a typical setup for running a walk-forward optimization in TradersStudio.

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

WEALTH-LAB: FEBRUARY 2014

The asymmetrical Donchian channel breakout system discussed in Oscar Cagigas's article in this issue ("The Degree Of Complexity") for model purposes is already built into Wealth-Lab. Thus, we don't need to provide code this time around. The feature is called Channel Breakout VT and can be found under the *breakouts* folder in the "open strategy" dialog (Ctrl-O). Traders can control the channel periods interactively through strategy parameter sliders.

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

BACK TO LIST

NINJATRADER: FEBRUARY 2014

The Donchian trading system, as discussed by Oscar Cagigas in his article in this issue, "The Degree Of Complexity," has been implemented as a strategy available for download at www.ninjatrader.com/SC/February2014SC.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 DonchianTradingSystem file.

A sample chart implementing the strategy is shown in Figure 8.

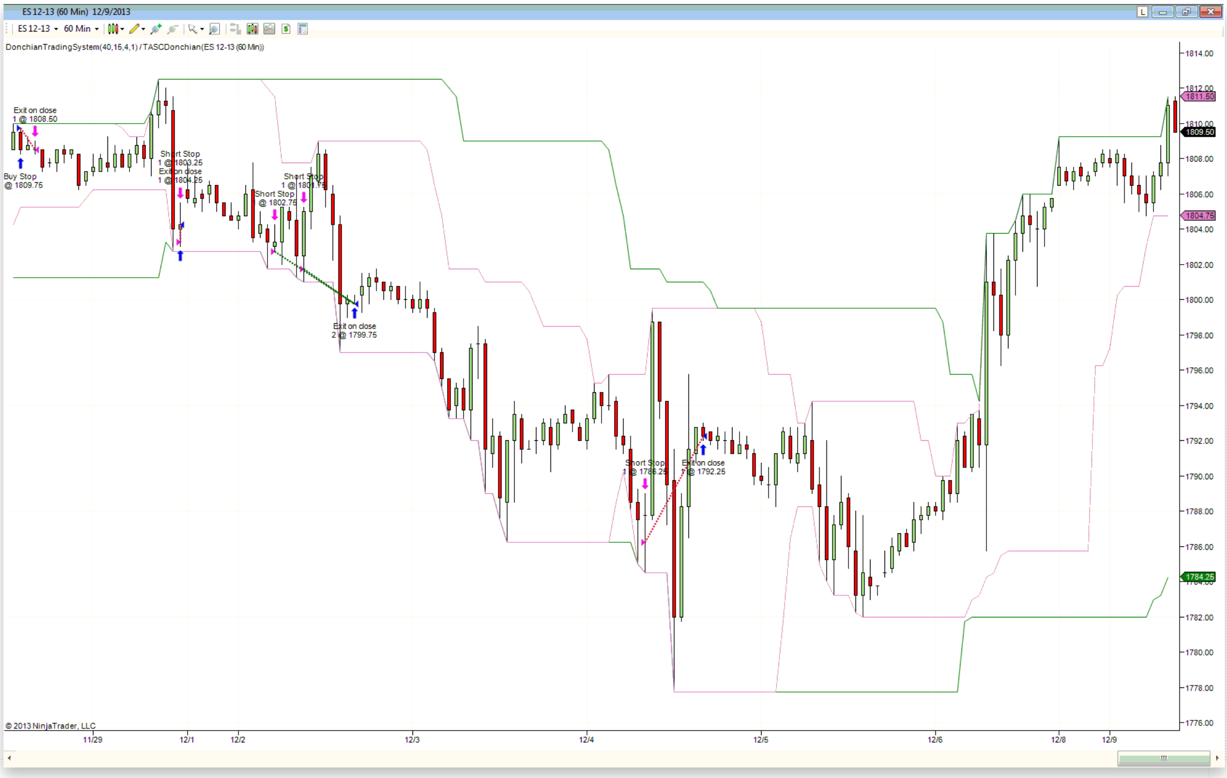


FIGURE 8: NINJATRADER. This screenshot shows the Donchian trading system applied to a 60-minute chart of the emini S&P 500. The lines represent the longer Donchian channel and slower Donchian channel optimized at 40 and 15, respectively.

—Raymond Deux & Cal Hueber
NinjaTrader, LLC
www.ninjatrader.com

BACK TO LIST

UPDATA: FEBRUARY 2014

In "The Degree Of Complexity" in this issue, author Oscar Cagigas develops systems with increasingly larger numbers of factors to show the effect of backtested performance dropoff between in- and out-of-sample data. The example given in Figure 9 is of a Donchian breakout system with a volatility filter and a trailing stop exit that are both based on ATR.

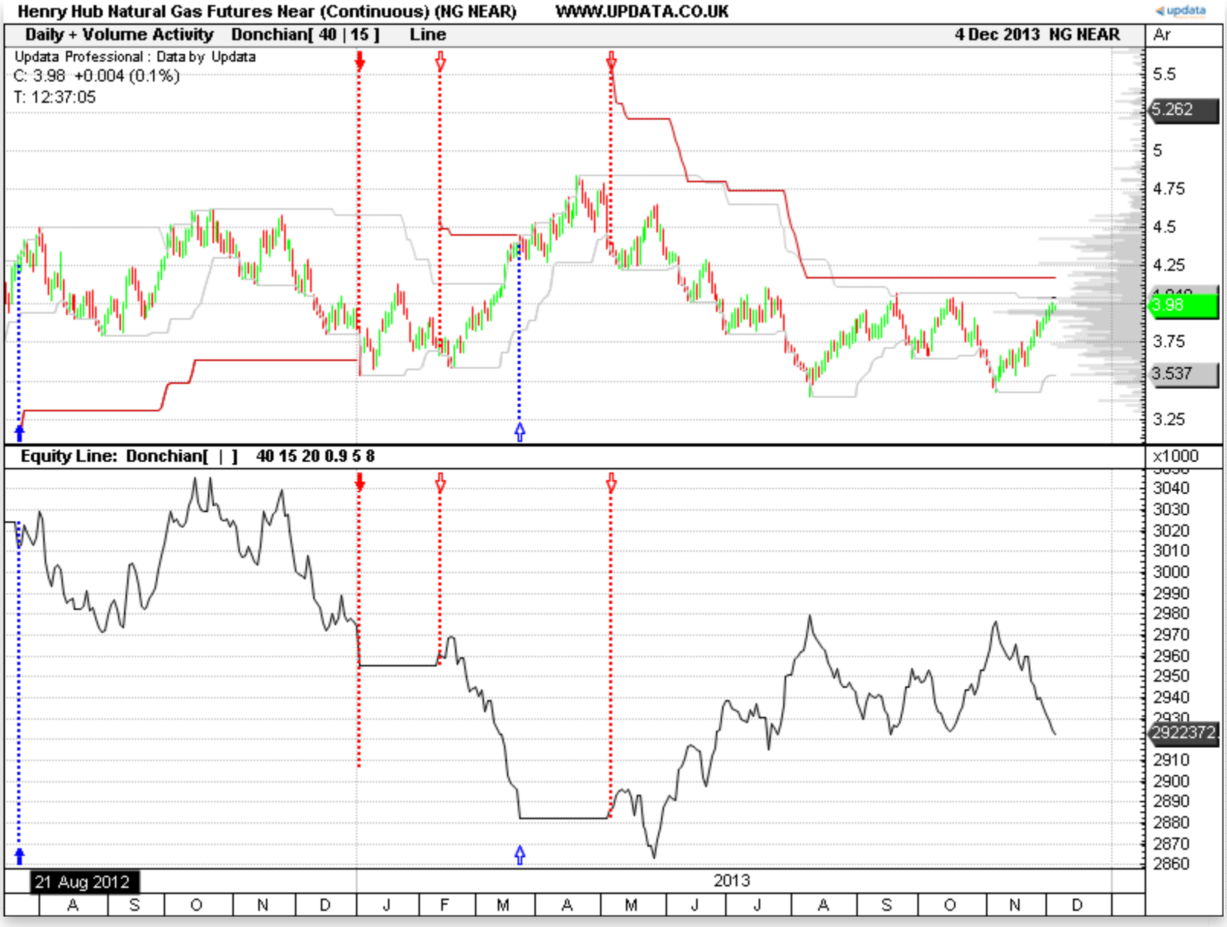


FIGURE 9: UPDATA. This chart shows the Donchian ATR filter system as applied to NYMEX-listed natural gas futures in daily resolution.

The Updata code for this 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 a firewall may paste the code below into the Updata custom editor and save it.

```
PARAMETER "Donchian Upper" #DONCHIANPeriodHigh=40
PARAMETER "Donchian Lower" #DONCHIANPeriodLow=15
PARAMETER "ATR Volatility Period" #ATRPeriodVol=20
PARAMETER "ATR Mult. 1" @K1=0.9
PARAMETER "ATR Stop Period" #ATRPeriodStop=5
PARAMETER "ATR Mult. 2" @K2=0.8
@ATRStopLong=0
@ATRStopShort=0
@DonchianUpper=0
@DonchianLower=0
DISPLAYSTYLE 4LINES
INDICATORTYPE TOOL
PLOTSTYLE LINE RGB(200,200,200)
PLOTSTYLE2 LINE RGB(200,200,200)
PLOTSTYLE3 LINE RGB(200,0,0)
PLOTSTYLE4 LINE RGB(200,0,0)
```

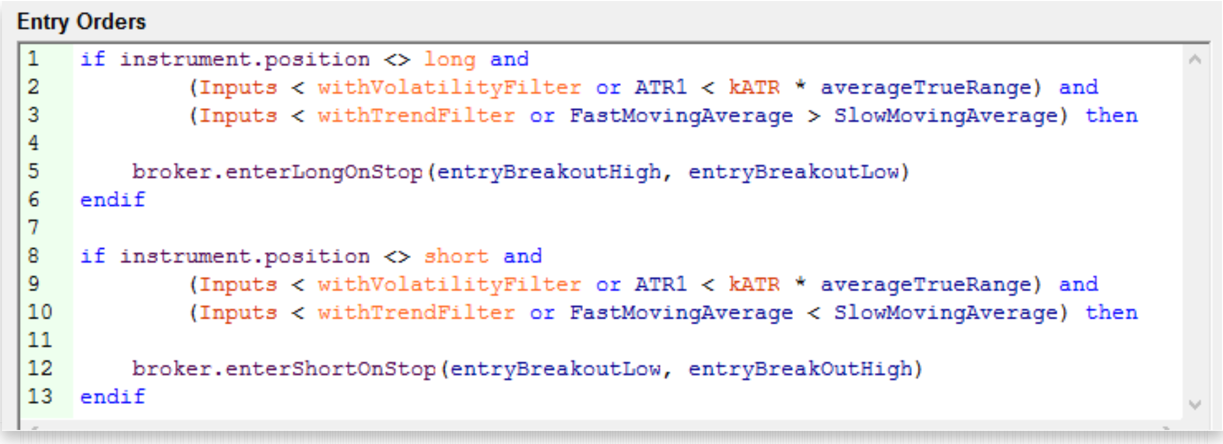
```
NAME "Donchian[" #DONCHIANPeriodHigh "|" #DONCHIANPeriodLow "]" ""
FOR #CURDATE=#DONCHIANPeriodHigh TO #LASTDATE
   @DonchianUpper=PHIGH(HIGH(1),#DONCHIANPeriodHigh)
   @DonchianLower=PLOW(LOW(1),#DONCHIANPeriodLow)
   'STOP EXIT
   IF ORDERISOPEN=1 AND LOW<@ATRStopLong
      SELL @ATRStopLong
   ELSEIF ORDERISOPEN=-1 AND HIGH>@ATRStopShort
      COVER @ATRStopShort
   ENDIF
   'DONCHIAN ENTRIES WITH ATR FILTER
   IF HIGH>@DonchianUpper AND ATR(1)<@K1*ATR(#ATRPeriodVol) AND ORDERISOPEN=0
      BUY CLOSE
      @ATRStopLong=CLOSE-@K2*ATR(#ATRPeriodStop)
   ELSEIF LOW<@DonchianLower AND ATR(1)<@K1*ATR(#ATRPeriodVol) AND ORDERISOPEN=0
      SHORT CLOSE
      @ATRStopShort=CLOSE+@K2*ATR(#ATRPeriodStop)
   ENDIF
   'TRAILING STOP
   IF ORDERISOPEN=1
      @ATRStopLong=MAX(CLOSE-@K2*ATR(#ATRPeriodStop),@ATRStopLong)
      @PLOT3=@ATRStopLong
      @PLOT4=-10000
   ELSEIF ORDERISOPEN=-1
      @ATRStopShort=MIN(CLOSE+@K2*ATR(#ATRPeriodStop),@ATRStopShort)
      @PLOT3=-10000
      @PLOT4=@ATRStopShort
   ELSE
      @PLOT3=-10000
      @PLOT4=-10000
   ENDIF
   @PLOT=@DonchianUpper
   @PLOT2=@DonchianLower
NEXT
```

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

BACK TO LIST

TRADING BLOX: FEBRUARY 2014

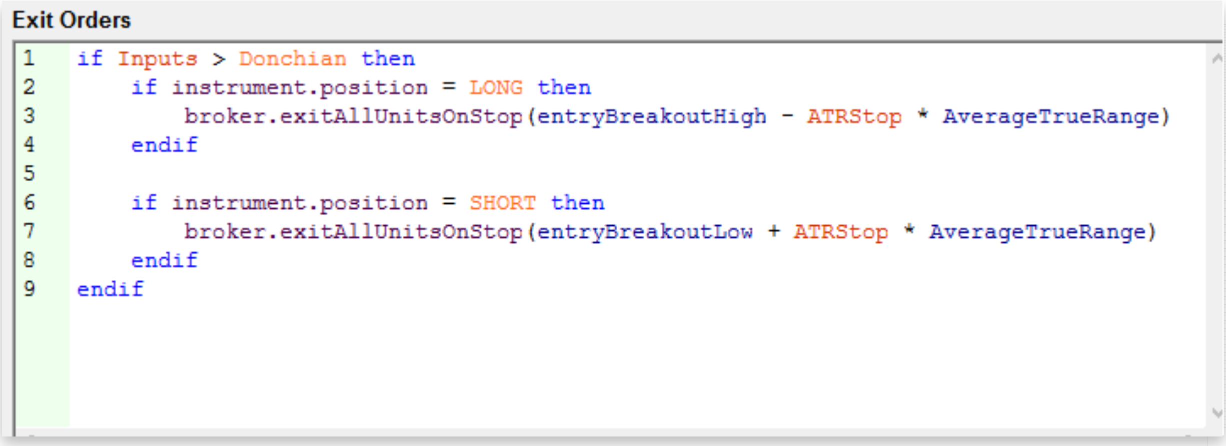
In "The Degree Of Complexity" in this issue, author Oscar Cagigas presents a Donchian breakout system with multiple sets of rules and filters. We are providing code for Trading Blox that includes the Donchian system rules (Figures 10 & 11) plus the ATR stop, volatility filter, and trend filter.



Entry Orders

```
if instrument.position <> long and
        (Inputs < withVolatilityFilter or ATR1 < kATR * averageTrueRange) and
        (Inputs < withTrendFilter or FastMovingAverage > SlowMovingAverage) then

    broker.enterLongOnStop(entryBreakoutHigh, entryBreakoutLow)
endif

if instrument.position <> short and
        (Inputs < withVolatilityFilter or ATR1 < kATR * averageTrueRange) and
        (Inputs < withTrendFilter or FastMovingAverage < SlowMovingAverage) then

    broker.enterShortOnStop(entryBreakoutLow, entryBreakOutHigh)
endif
```

FIGURE 10: TRADING BLOX, ENTRY RULES. Here are the entry rules for the Donchian system based on Oscar Cagigas's article in this issue.

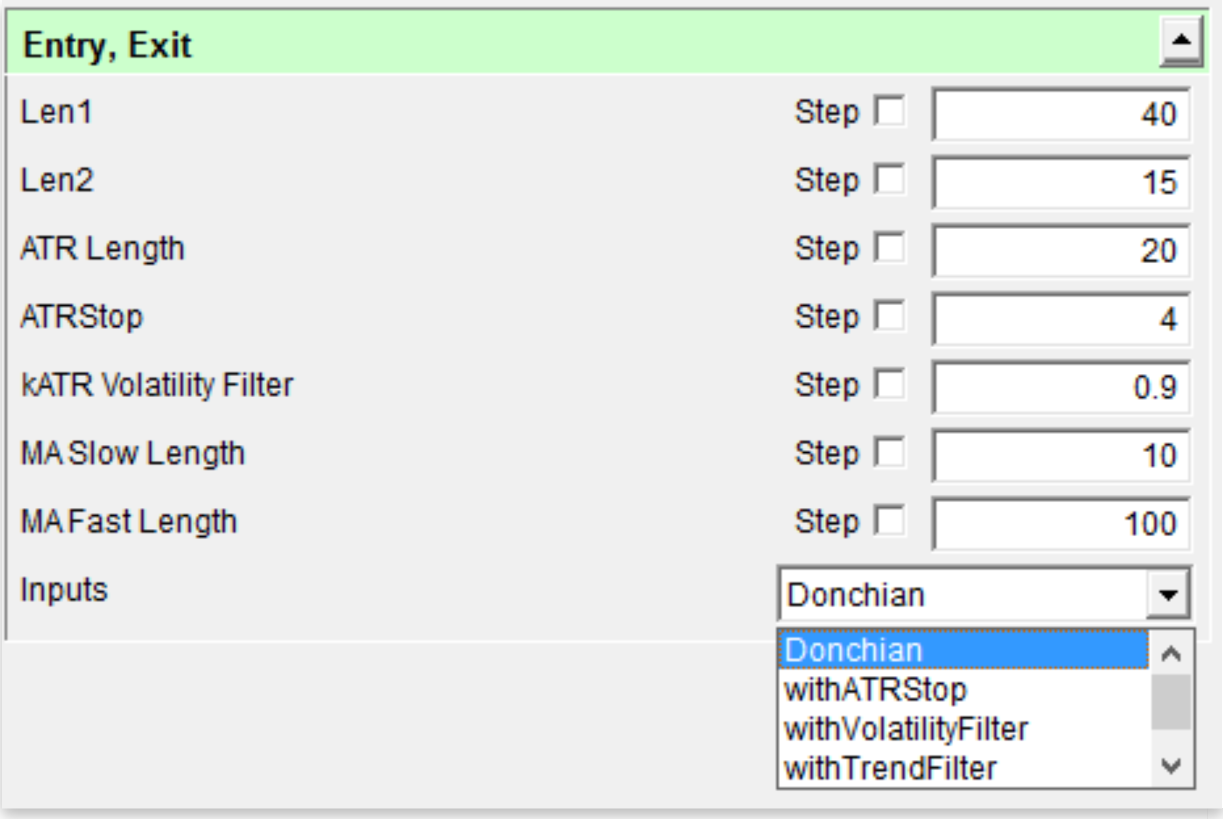


Exit Orders

```
if Inputs > Donchian then
    if instrument.position = LONG then
        broker.exitAllUnitsOnStop(entryBreakoutHigh - ATRStop * AverageTrueRange)
    endif

    if instrument.position = SHORT then
        broker.exitAllUnitsOnStop(entryBreakoutLow + ATRStop * AverageTrueRange)
    endif
endif
```

FIGURE 11: TRADING BLOX, EXIT RULES. Here are the exit rules.

The *inputs* parameter (Figure 12) allows each rule or filter to be selected, inclusive of the previous one, as in Cagigas's article.



Entry, Exit

Len1	Step ☐	40
Len2	Step ☐	15
ATR Length	Step ☐	20
ATRStop	Step ☐	4
kATR Volatility Filter	Step ☐	0.9
MA Slow Length	Step ☐	10
MA Fast Length	Step ☐	100
Inputs		Donchian

FIGURE 12: TRADING BLOX, RULE SELECTOR. The inputs parameter allows each rule or filter to be selected, inclusive of the previous one, as in Cagigas's article.

—Trading Blox
tradingblox.com

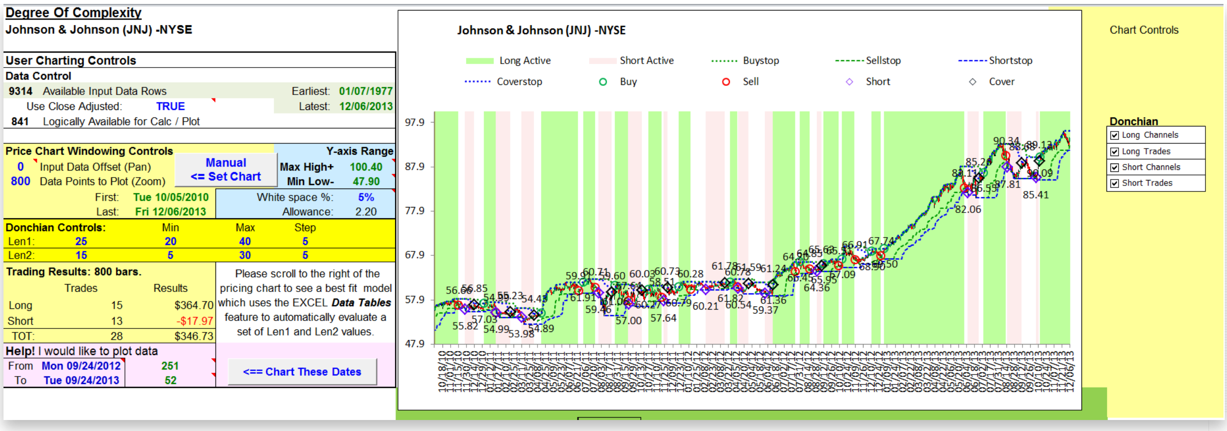
BACK TO LIST

MICROSOFT EXCEL: FEBRUARY 2014

Oscar Cagigas's article in this issue, "The Degree Of Complexity," investigates the beneficial and detrimental aspects of increasing the complexity of a trading system model in the quest to improve system performance through backtesting. Cagigas then measures the performance of the improved system model at each step by checking the results of the model on out-of-sample data.

The spreadsheet I am providing here demonstrates a version of his basic, two-parameter Donchian system that is the starting point for Cagigas's exploration of system complexity.

Using the built-in Excel capability for data tables (Figure 13), the spreadsheet I am providing incorporates a very simple modeling capability that cycles through *Len1* and *Len2* values using the low, high, and step size values for each, as found in the sidebar to Cagigas's article presenting AmiBroker code.



Degree Of Complexity
Johnson & Johnson (JNJ) -NYSE

User Charting Controls

Data Control

9314	Available Input Data Rows	Earliest:	01/07/1977
	Use Close Adjusted: TRUE	Latest:	12/06/2013
841	Logically Available for Calc / Plot		

Price Chart Windowing Controls

0 Input Data Offset (Pan)
800 Data Points to Plot (Zoom)
First: Tue 10/05/2010
Last: Fri 12/06/2013

Manual <= Set Chart

Y-axis Range
Max High+ 100.40
Min Low- 47.90
White space %: 5%
Allowance: 2.20

Donchian Controls:		Min	Max	Step
Len1:	25	20	40	5
Len2:	15	5	30	5

Trading Results: 800 bars.

	Trades	Results
Long	15	$364.70
Short	13	-$17.97
TOT:	28	$346.73

Please scroll to the right of the pricing chart to see a best fit model which uses the EXCEL ***Data Tables*** feature to automatically evaluate a set of Len1 and Len2 values.

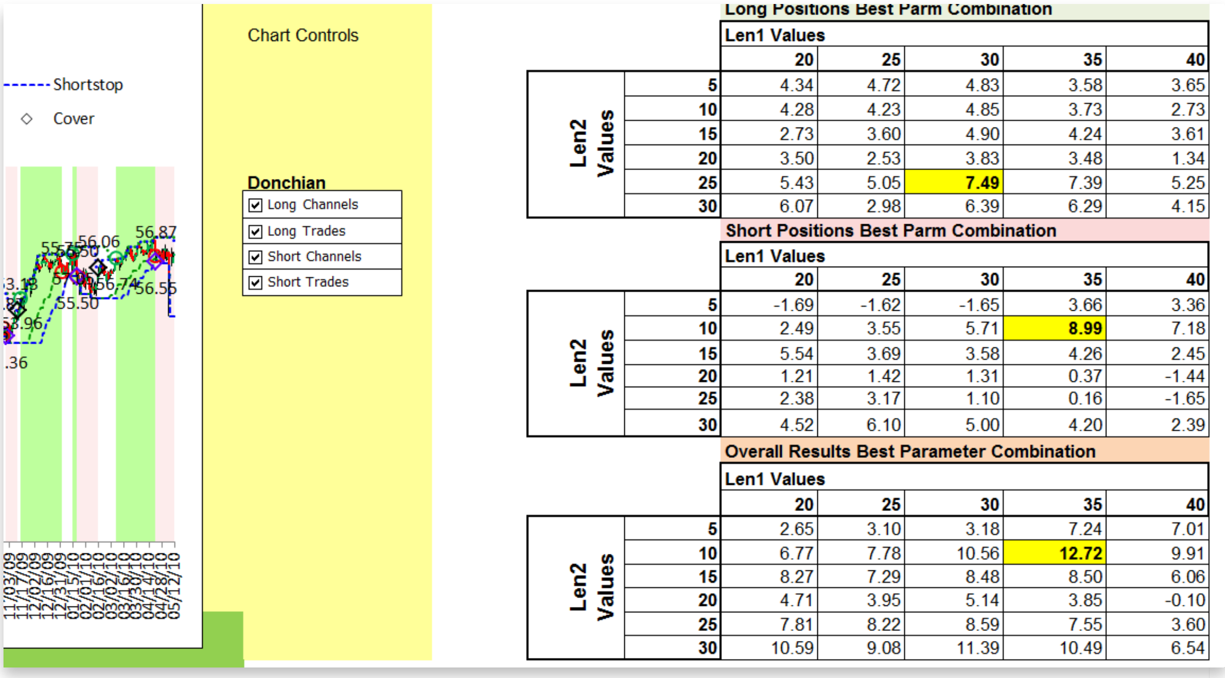
Help! I would like to plot data

From	Mon 09/24/2012	251
To	Tue 09/24/2013	52

<== Chart These Dates

FIGURE 13: EXCEL, DONCHIAN TRADING SYSTEM. This sample chart shows Johnson & Johnson in the simple Donchian system described in Oscar Cagigas's article in this issue, "The Degree Of Complexity."

You can perform a simple backtest by specifying an offset of something like 900 and using a "points to plot" value of 800. Then click the *manual set chart* button and wait for things to settle. When the flickering stops, scroll to the right of the pricing chart to see the simulation results (Figure 14).



Long Positions Best Parm Combination

Len2 Values	Len1 Values 20	25	30	35	40
5	4.34	4.72	4.83	3.58	3.65
10	4.28	4.23	4.85	3.73	2.73
15	2.73	3.60	4.90	4.24	3.61
20	3.50	2.53	3.83	3.48	1.34
25	5.43	5.05	**7.49**	7.39	5.25
30	6.07	2.98	6.39	6.29	4.15

Short Positions Best Parm Combination

Len2 Values	Len1 Values 20	25	30	35	40
5	-1.69	-1.62	-1.65	3.66	3.36
10	2.49	3.55	5.71	**8.99**	7.18
15	5.54	3.69	3.58	4.26	2.45
20	1.21	1.42	1.31	0.37	-1.44
25	2.38	3.17	1.10	0.16	-1.65
30	4.52	6.10	5.00	4.20	2.39

Overall Results Best Parameter Combination

Len2 Values	Len1 Values 20	25	30	35	40
5	2.65	3.10	3.18	7.24	7.01
10	6.77	7.78	10.56	**12.72**	9.91
15	8.27	7.29	8.48	8.50	6.06
20	4.71	3.95	5.14	3.85	-0.10
25	7.81	8.22	8.59	7.55	3.60
30	10.59	9.08	11.39	10.49	6.54

FIGURE 14: EXCEL, BEST-FIT USING OFFSET OF 900. Here is the best-fit backtest model at an offset of 900 using the built-in capability for Excel data tables.

Capture the best-fit *Len1* and *Len2* values from the appropriate data table and enter them in cells B16 and B17, respectively.

Out-of-sample testing may be simulated by setting the offset to zero, leaving the points to plot at 800.

Once again, click the *manual set chart* button and wait for things to settle. When the flickering stops, scroll to the right of the pricing chart to see just how much things have changed (Figure 15).

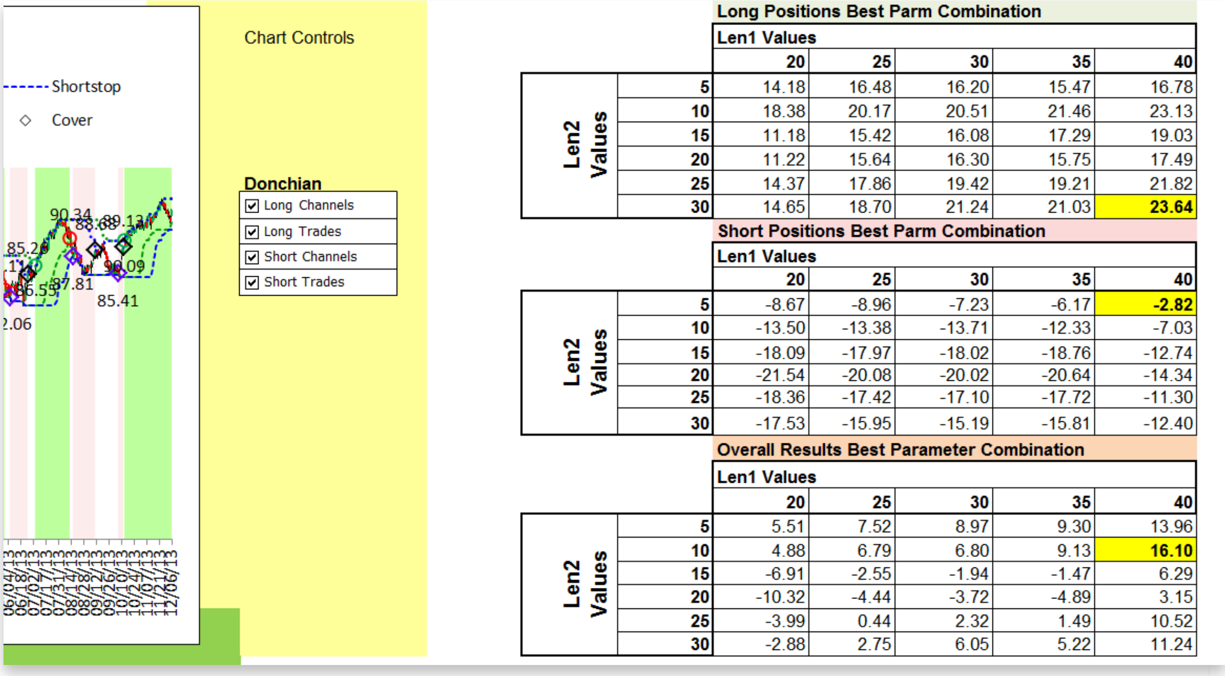


Chart Controls

Donchian
☑ Long Channels
☑ Long Trades
☑ Short Channels
☑ Short Trades

Long Positions Best Parm Combination

Len2 Values	Len1 Values 20	25	30	35	40
5	14.18	16.48	16.20	15.47	16.78
10	18.38	20.17	20.51	21.46	23.13
15	11.18	15.42	16.08	17.29	19.03
20	11.22	15.64	16.30	15.75	17.49
25	14.37	17.86	19.42	19.21	21.82
30	14.65	18.70	21.24	21.03	**23.64**

Short Positions Best Parm Combination

Len2 Values	Len1 Values 20	25	30	35	40
5	-8.67	-8.96	-7.23	-6.17	**-2.82**
10	-13.50	-13.38	-13.71	-12.33	-7.03
15	-18.09	-17.97	-18.02	-18.76	-12.74
20	-21.54	-20.08	-20.02	-20.64	-14.34
25	-18.36	-17.42	-17.10	-17.72	-11.30
30	-17.53	-15.95	-15.19	-15.81	-12.40

Overall Results Best Parameter Combination

Len2 Values	Len1 Values 20	25	30	35	40
5	5.51	7.52	8.97	9.30	13.96
10	4.88	6.79	6.80	9.13	**16.10**
15	-6.91	-2.55	-1.94	-1.47	6.29
20	-10.32	-4.44	-3.72	-4.89	3.15
25	-3.99	0.44	2.32	1.49	10.52
30	-2.88	2.75	6.05	5.22	11.24

FIGURE 15: EXCEL, BEST-FIT USING OFFSET OF ZERO. Here is the best-fit “out-of-sample” using an offset of zero.

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” 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 February 2014 issue of
Technical Analysis of STOCKS & COMMODITIES magazine.

TRADERS' TIPS

March 2014

For this month's Traders' Tips, the focus is Perry Kaufman's article in this issue, "Timing The Market With Pairs Logic." Here we present the March 2014 Traders' Tips code with possible implementations in various software.

Code for TradeStation is already provided in Kaufman's article. Subscribers will find that code at the Subscriber Area of our website. Presented here is an overview of possible implementations for other software.

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

TRADESTATION: MARCH 2014

In "Timing The Market With Pairs Logic" in this issue, author Perry Kaufman discusses a trading system that buys and sells stocks when they are oversold and overbought relative to an index. The author has supplied the TradeStation EasyLanguage strategy code as well as the required custom function mentioned in the article. We have additionally created an indicator named PJK_TSMStress based on the author's function, to display the stress level as shown in Figure 1 of Kaufman's article. In addition to backtesting the strategy in a TradeStation chart, remember that you can use TradeStation's Portfolio Maestro product to quickly backtest on a portfolio of symbols of your choice.

Following is the EasyLanguage code for the PJK_TSMStress indicator:

```
{ Based on PJK_Stress function
Copyright 2013, P.J.Kaufman. All rights reserved. }
inputs:
        Period( 60 ),
        OBLevel( 90 ),
        OSLevel( 10 ),
        NormalLevel( 50 ) ;
variables:
        Stoch1( 0 ),
        Stoch2( 0 ),
        Diff( 0 ),
        Range1( 0 ),
        Range2( 0 ),
        StressValue( 0 ) ;

StressValue = 50 ;
Stoch1 = 50 ;
Stoch2 = 50 ;

{ raw stochastics for price1 and price2 }
Range1 = Highest( High, Period )
        - Lowest( Low, Period ) ;
Range2 = Highest( High of Data2, Period)
        - Lowest( Low of Data2, Period ) ;

if Range1 <> 0 and Range2 <> 0 then
        begin
        Stoch1 = ( Close
                - Lowest( Low, Period ) ) / Range1 ;
        Stoch2 = ( Close of Data2
                - Lowest( Low of Data2, Period ) ) / Range2 ;

        { difference in stochastics }
        Diff = Stoch1 - Stoch2;
        { stress indicator }
        Range1 = Highest( Diff, Period )
                - Lowest( Diff, Period ) ;
        if Range1 <> 0 then
                StressValue = 100 * ( Diff
                        - Lowest( Diff, Period ) ) / Range1 ;
        end ;

Plot1( StressValue, "Stress" ) ;
Plot2( Stoch1 * 100, "D1 Stoch" ) ;
Plot3( Stoch2 * 100, "D2 Stoch" ) ;
Plot4( OBLevel, "OverBought" ) ;
Plot5( OSLevel, "OverSold" ) ;
Plot6( NormalLevel, "Normal" ) ;
```

To download the EasyLanguage code, please visit our TradeStation & EasyLanguage support forum. The code can be found at http://www.tradestation.com/TASC-2014. The ELD filename is "_TASC_PJK_PAIRS.ELD."

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

A sample chart showing the PJK_TSMStress indicator is shown in Figure 1.

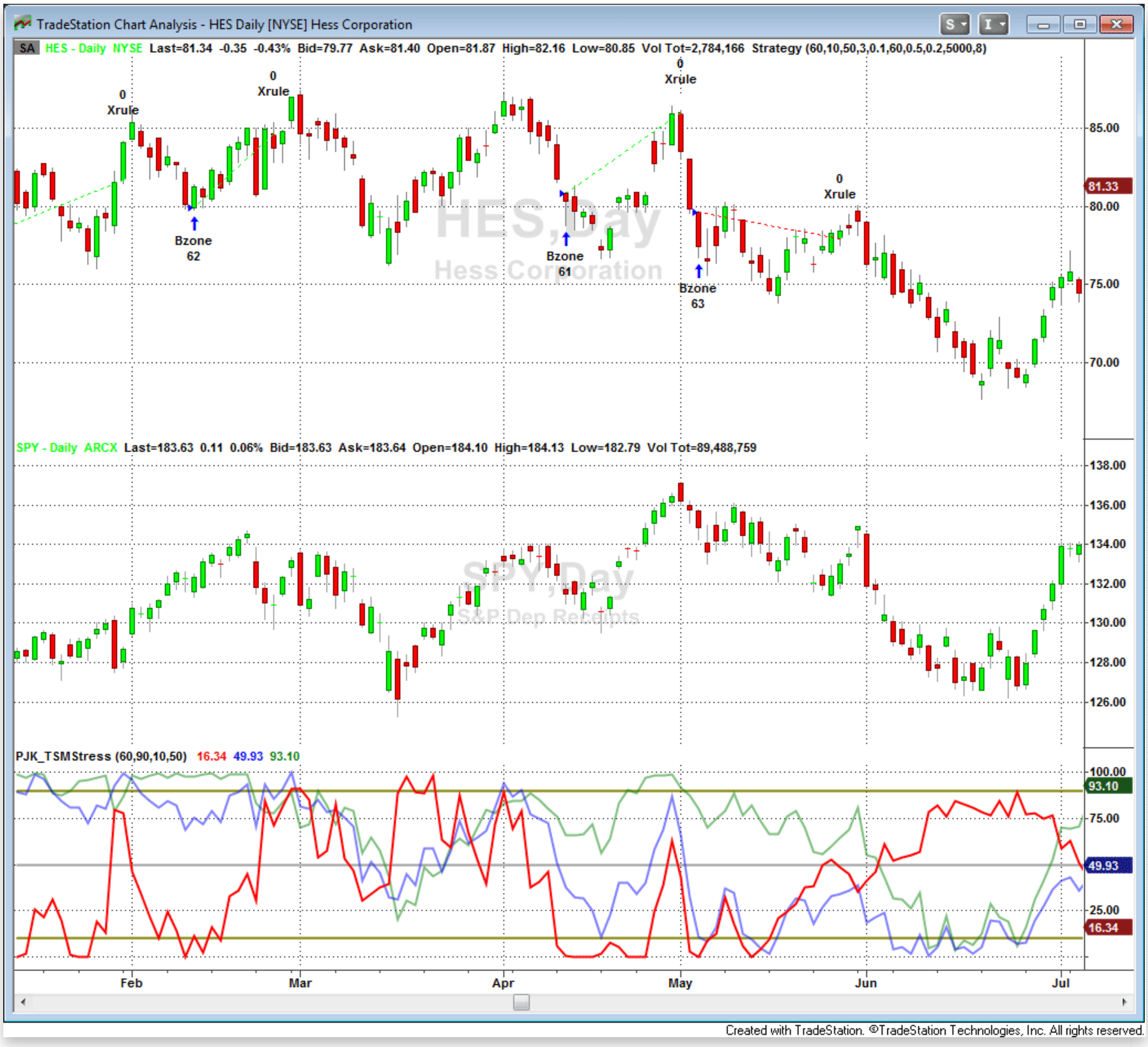


FIGURE 1: TRADESTATION. Here is a daily chart of Hess (HES) with the indicator and strategy applied.

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

CQG: MARCH 2014

For this month’s Traders’ Tip, we’re providing CQG code for the stress function based on Perry Kaufman’s article in this issue, “Timing The Market With Pairs Logic.”

CQG code for the study:

```
/*Stress Function, P.J. Kaufman*/
/*Raw Stochastics for Price1 and Price2*/
Data2:= S.US.SPY;
Range1:= HiLevel(@,period,0) - LoLevel(@,period,0);
Range2:= HiLevel(Data2,period,0) - LoLevel(Data2,period,0);
Stoch1:= (Close(@) - LoLevel(@,period,0))/Range1;
Stoch2:= (Data2 - LoLevel(Data2,period,0))/Range2;

/*Difference in Stochastics*/
Diff:= Stoch1-Stoch2;

/*Stress Indicator*/
Range1S:= HiLevel(Diff,period,0) - LoLevel(Diff,period,0);
StressValue:= 100*(Diff - LoLevel(Diff,period,0))/Range1S;
```

The study has one parameter, *period*, which may be configured in the “modify study parameters” window after the study has been applied to a chart in CQG. An example of the study applied to Hess (HES) is depicted in the chart shown in Figure 2.

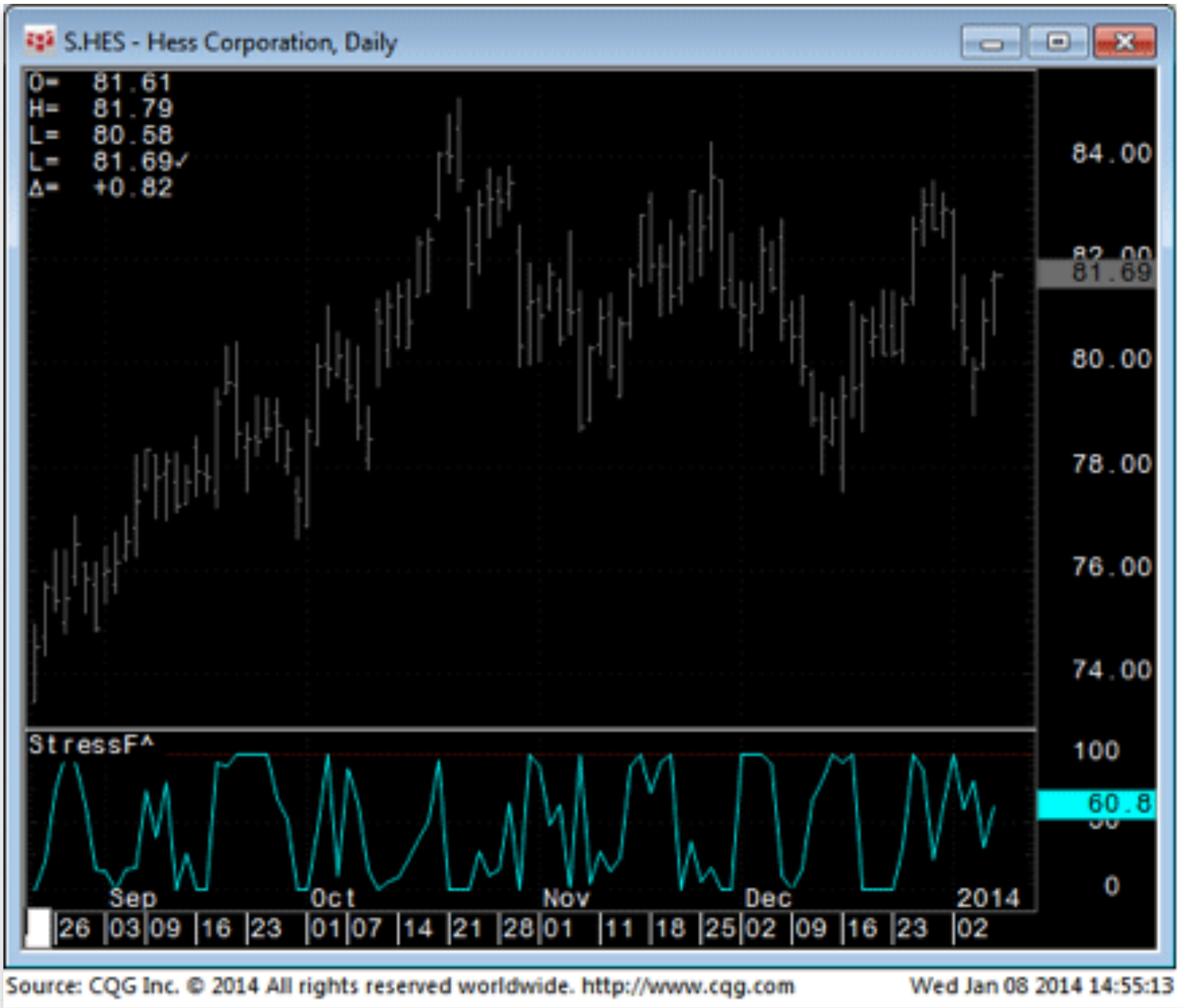


FIGURE 2: CQG. Here is an example of the study using Hess Corp. (HES).

To discuss this study or download a component PAC that includes complete formula code, please visit CQG Forums and CQG Workspaces. Our team of expert product specialists can advise you on the usage, application, and code for the study.

Trading and investment carry a high level of risk, and CQG, Inc. does not make any recommendations for buying or selling any financial instruments. We offer educational information on ways to use CQG trading tools, but it is up to our customers and other readers to make their own trading and investment decisions or to consult with a registered investment advisor.

—CQG, Inc.
www.CQG.com

BACK TO LIST

METASTOCK: MARCH 2014

Perry Kaufman's article in this issue, "Timing The Market With Pairs Logic," describes his stress indicator and how to use it in pair trading. The MetaStock code for this indicator based on his article is shown here:

```
Stress:

s1c:= Security("SPY",C);
s1l:= Security("SPY",L);
s1h:= Security("SPY",H);
Prd:= Input("Stoch Periods", 1, 100, 10);
R1:=(HHV(H,prd)-LLV(L,prd));
r2:= (HHV(s1h,prd)-LLV(s1l,prd));
ST1:= (C-LLV(L,prd))/r1;
ST2:= (s1c-LLV(s1l,prd))/r2;
diff:= ST1-ST2;
Sr1 := HHV(diff,prd)-LLV(diff,prd);
Stress := If(SR1<>0,100*((diff-LLV(diff,prd))/SR1), PREV);
stress;
100 * ST1;
100 * ST2
```

—William Golson
MetaStock Technical Support
www.metastock.com

BACK TO LIST

THINKORSWIM: MARCH 2014

In "Timing The Market With Pairs Logic" in this issue, author Perry Kaufman explains how to backtest the idea of hedging with an index-based ETF. Based on his article, we have created two new strategies and a new study for thinkorswim users in our proprietary scripting language, thinkScript. One strategy is for the equity and the other strategy is for the ETF.

A sample chart is shown in Figure 3.

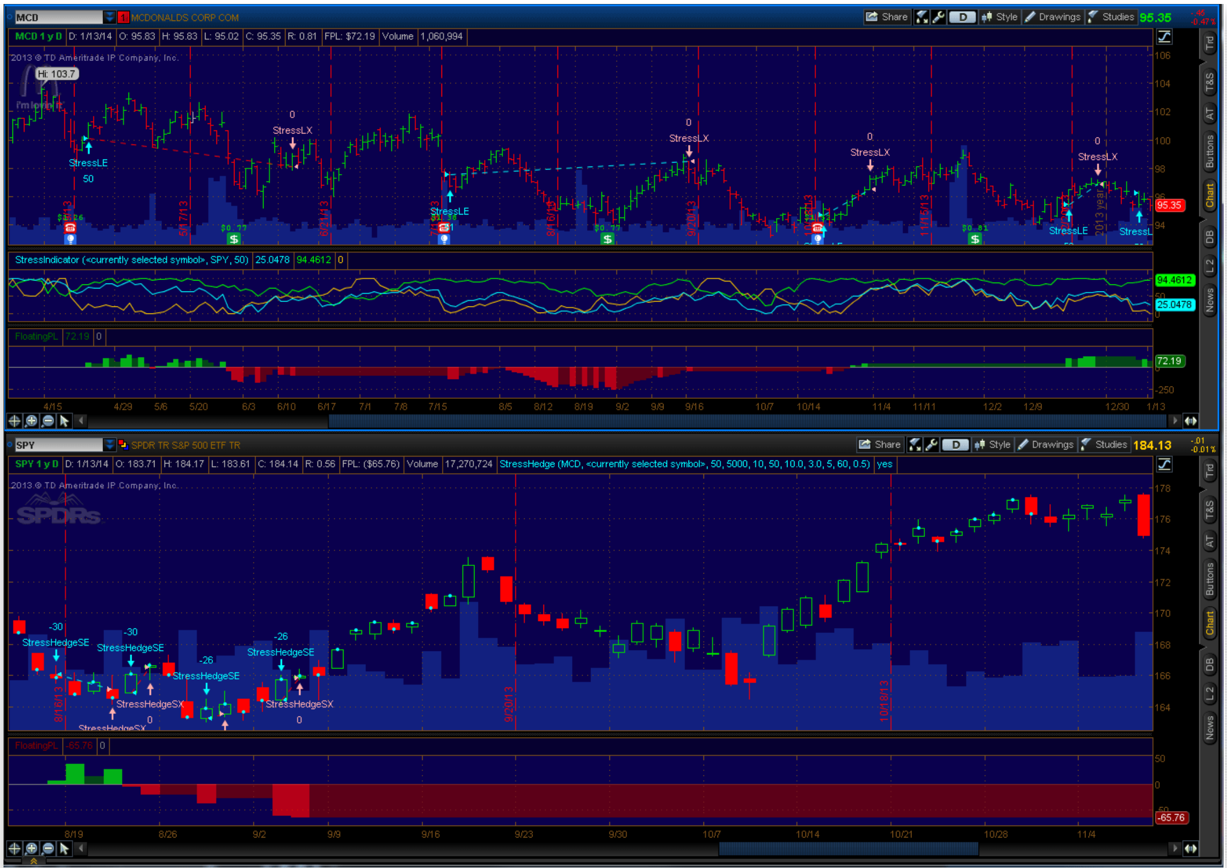


FIGURE 3: THINKORSWIM

For the Equity Strategy click here or:

1. From our TOS Charts, Select **Studies** → **Edit Studies**.
2. Select the **Strategy** tab in the upper left hand corner.
3. Select **New** in the lower left hand corner.
4. Name the strategy (i.e. Stress)
5. Click in the script editor window, remove “addOrder(OrderType.BUY_AUTO, no);” and paste the following:

```
script StressIndicator {
    input stock = "<currently selected symbol>";
    input index = "SPY";
    input length = 50;

    def rangeStock = Highest(high(stock), length) - Lowest(low(stock), length);
    plot StochStock = 100 * if rangeStock != 0 then (close(stock) -
Lowest(low(stock), length)) / rangeStock else 0;

    def rangeIndex = Highest(high(index), length) - Lowest(low(index), length);
    plot StochIndex = 100 * if rangeIndex != 0 then (close(index) -
Lowest(low(index), length)) / rangeIndex else 0;

    def diff = StochStock - StochIndex;
    def rangeDiff = Highest(diff, length) - Lowest(diff, length);
    plot Stress = 100 * if rangeDiff != 0 then (diff - Lowest(diff, length)) /
rangeDiff else 0;
}

input index = "SPY";
input length = 50;
input investment = 5000;
input entryLevel = 10;
input exitLevel = 50;
input stopLoss = 10.0;
input minPrice = 3.0;
input minPriceLength = 5;

def minPirceOk = Lowest(close, minPriceLength) > minPrice;
def stress = reference StressIndicator(index = index, length = length).Stress;
def entryPrice = EntryPrice();
def crisisStop = close / entryPrice - 1 < -stopLoss / 100;
def isReady = if !isReady[1] and stress > 50 then yes else if IsNaN(crisisStop)
then isReady[1] else if !IsNaN(entryPrice) and crisisStop then no else
isReady[1];

AddOrder(OrderType.BUY_TO_OPEN, isReady and stress <= entryLevel and minPirceOk,
tradeSize = investment / close, tickColor = GetColor(1), arrowColor =
GetColor(1), name = "StressLE");

AddOrder(OrderType.SELL_TO_CLOSE, !minPirceOk, tickColor = GetColor(2),
arrowColor = GetColor(2), name = "StressMinPriceLX");
AddOrder(OrderType.SELL_TO_CLOSE, stress >= exitLevel, tickColor = GetColor(2),
arrowColor = GetColor(2),  name = "StressLX");
AddOrder(OrderType.SELL_TO_CLOSE, crisisStop, tickColor = GetColor(2),
arrowColor = GetColor(2), name = "StressCrisisStopLX");
```

For the ETF Strategy click here or:

1. From our TOS Charts, Select **Studies** → **Edit Studies**.
2. Select the **Strategy** tab in the upper left hand corner.
3. Select **New** in the lower left hand corner.
4. Name the strategy (i.e. StressHedge)
5. Click in the script editor window, remove “addOrder(OrderType.BUY_AUTO, no);” and paste the following:

```
script StressIndicator {
    input stock = "<currently selected symbol>";
    input index = "SPY";
    input length = 50;

    def rangeStock = Highest(high(stock), length) - Lowest(low(stock), length);
    plot StochStock = 100 * if rangeStock != 0 then (close(stock) -
Lowest(low(stock), length)) / rangeStock else 0;

    def rangeIndex = Highest(high(index), length) - Lowest(low(index), length);
    plot StochIndex = 100 * if rangeIndex != 0 then (close(index) -
Lowest(low(index), length)) / rangeIndex else 0;

    def diff = StochStock - StochIndex;
    def rangeDiff = Highest(diff, length) - Lowest(diff, length);
    plot Stress = 100 * if rangeDiff != 0 then (diff - Lowest(diff, length)) /
rangeDiff else 0;
}

input stock = "AAPL";
input index = "<currently selected symbol>";
input length = 50;
input investment = 5000;
input entryLevel = 10;
input exitLevel = 50;
input stopLoss = 10.0;
input minPrice = 3.0;
input minPriceLength = 5;
input hedgeLength = 60;
input hedgeRatio = 0.5;

def minPirceOk = Lowest(close(stock), minPriceLength) > minPrice;
def stress = reference StressIndicator(stock = stock, index = index, length =
length).Stress;

def stockPosition = {default none, long};
def entryPrice;
def stockSize;
def crisisStop = close(stock) / entryPrice[1] - 1 < -stopLoss / 100;
def isReady = if !isReady[1] and stress > 50 then yes else if IsNaN(crisisStop)
then isReady[1] else if !IsNaN(entryPrice[1]) and crisisStop then no else
isReady[1];

def buy = isReady and stress <= entryLevel and minPirceOk;
def sell = !minPirceOk or stress >= exitLevel or crisisStop;

if buy {
    stockPosition = stockPosition.long;
    entryPrice = open(stock)[-1];
    stockSize = investment / close(stock)[-1];
} else if sell {
    stockPosition = stockPosition.none;
    entryPrice = Double.NaN;
    stockSize = 0;
} else {
    stockPosition = stockPosition[1];
    entryPrice = CompoundValue(1, entryPrice[1], Double.NaN);
    stockSize = stockSize[1];
}
def stockChange = close(stock) / close(stock)[1] - 1;
def indexChange = close(index) / close(index)[1] - 1;
def indexTrend = Average(close(index), hedgeLength);
def stockVol = StDev(stockChange, hedgeLength);
def indexVol = StDev(indexChange, hedgeLength);
def indexSize = stockSize * hedgeRatio * stockVol / indexVol;

plot Position = stockPosition;
Position.SetPaintingStrategy(PaintingStrategy.BOOLEAN_POINTS);

AddOrder(OrderType.SELL_TO_OPEN, stockPosition == stockPosition.long and
indexTrend < indexTrend[1], tradeSize = indexSize, tickColor = GetColor(1),
arrowColor = GetColor(1), name = "StressHedgeSE");
AddOrder(OrderType.BUY_TO_CLOSE, stockPosition == stockPosition.none or
indexTrend > indexTrend[1], tickColor = GetColor(2), arrowColor = GetColor(2),
name = "StressHedgeSX");
```

For the Study click here or:

1. From our TOS Charts, Select **Studies** → **Edit Studies**.
2. Select the **Studies** tab in the upper left hand corner.
3. Select **New** in the lower left hand corner.
4. Name the strategy (i.e. StressIndicator)
5. Click in the script editor window, remove “plot Data = close;” and paste the following:

```
declare lower;

input stock = "<currently selected symbol>";
input index = "SPY";
input length = 50;

def rangeStock = Highest(high(stock), length) - Lowest(low(stock), length);
plot StochStock = 100 * if rangeStock != 0 then (close(stock) -
Lowest(low(stock), length)) / rangeStock else 0;

def rangeIndex = Highest(high(index), length) - Lowest(low(index), length);
plot StochIndex = 100 * if rangeIndex != 0 then (close(index) -
Lowest(low(index), length)) / rangeIndex else 0;

def diff = StochStock - StochIndex;
def rangeDiff = Highest(diff, length) - Lowest(diff, length);
plot Stress = 100 * if rangeDiff != 0 then (diff - Lowest(diff, length)) /
rangeDiff else 0;

StochStock.SetDefaultColor(GetColor(1));
StochIndex.SetDefaultColor(GetColor(6));
Stress.SetDefaultColor(GetColor(4));
```

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

BACK TO LIST

WEALTH-LAB: MARCH 2014

In this issue, Perry Kaufman’s article “Timing The Market With Pairs Logic” promises an interesting new take on pair trading. As discussed in the article, combining Kaufman’s stochastic-derived intermarket *stress indicator* with a few clear position-sizing and risk-management rules lays the foundation for a long-only market timing system.

To execute the trading system that we are presenting here in Wealth-Lab strategy code, Wealth-Lab users need to install (or update to) the latest version of our TASCIndicators library from the *extensions* section of our website if they haven’t already done so, and restart Wealth-Lab.

A sample chart showing the stress indicator on Hess is in Figure 4.

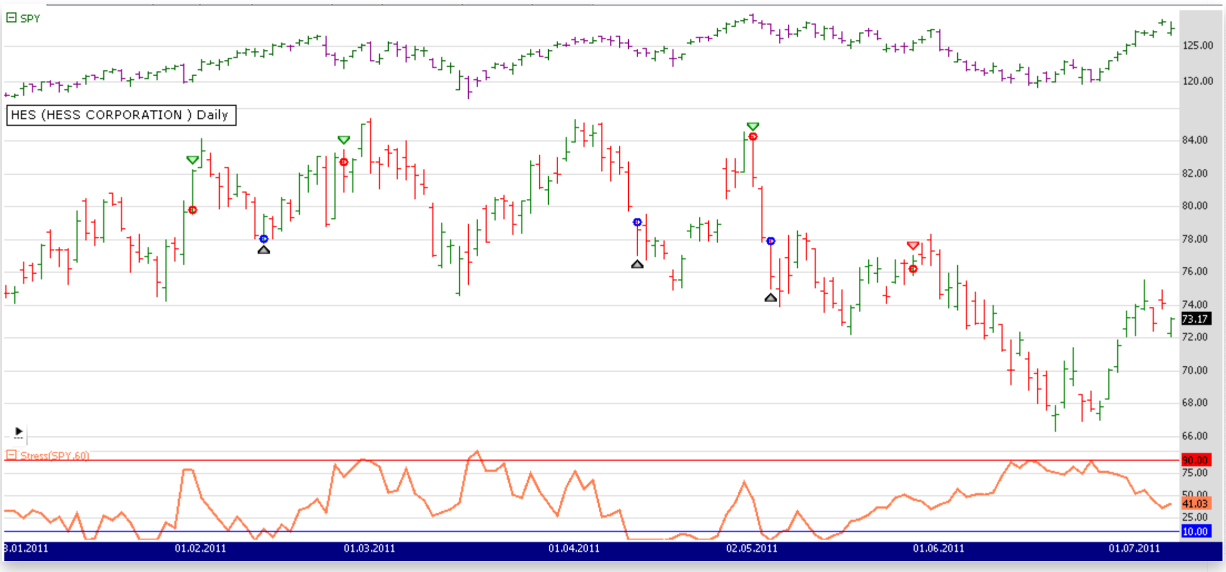


FIGURE 4: WEALTH-LAB. Here is a sample Wealth-Lab 6 chart illustrating application of the system's rules on a daily chart of HES (middle pane). An SPY chart is shown in the upper pane, and the stress indicator is plotted in the bottom pane.

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 PJKPairs1 : WealthScript
        {
                private StrategyParameter paramPeriod;
                private StrategyParameter paramBuy;
                private StrategyParameter paramSell;

                public PJKPairs1()
                {
                        paramPeriod = CreateParameter("Period", 60, 10, 100, 10);
                        paramBuy = CreateParameter("Buy", 10, 10, 30, 10);
                        paramSell = CreateParameter("Sell", 50, 50, 90, 10);
                }

                protected override void Execute()
                {
                        int period = paramPeriod.ValueInt;
                        int buy = paramBuy.ValueInt;
                        int sell = paramSell.ValueInt;

                        string stock = Bars.Symbol;
                        string idx = "SPY";
                        Bars index = GetExternalSymbol( idx, true );
                        DataSeries indexTrend = SMA.Series( index.Close, period );
                        DataSeries stress = Stress.Series( Bars, index, period );

                        ChartPane sPane = CreatePane( 30, false, true );
                        PlotSeries( sPane, stress, Color.Coral, LineStyle.Solid, 2 );
                        DrawHorzLine( sPane, 10, Color.Blue, LineStyle.Solid, 1 );
                        DrawHorzLine( sPane, 90, Color.Red, LineStyle.Solid, 1 );

                        ChartPane idxPane = CreatePane( 30, true, true );
                        PlotSymbol( idxPane, index, Color.DarkGreen,
Color.DarkMagenta );

                        HideVolume();

                        for(int bar = Bars.FirstActualBar + period; bar < Bars.Count;
bar++)
                        {
                                List<Position> lst = new List<Position>();
                                lst.AddRange(Positions);

                                if( SymbolIsActive(stock) )
                                {
                                        if( stress[bar] >= sell )
                                        {
                                                int lastActivePositionInStock =
LastActivePositionInSym(lst,stock);

                                                if( (stress[bar] >= sell) &&
lastActivePositionInStock > -1 )
                                                        SellAtMarket( bar+1,
Positions[lastActivePositionInStock], "Xrule" );
                                        }

                                        if( SymbolIsActive(idx))
                                        {
                                                if( indexTrend[bar] <
indexTrend[bar-1] )
                                                {
                                                        int lastActivePositionInIndex
= LastActivePositionInSym(lst,idx);

                                                        if( lastActivePositionInIndex
> -1 )
                                                        {
                                                                SetContext( idx, true
);
                                                                SellAtMarket( bar+1,
Positions[lastActivePositionInIndex], "Sell " + idx );
                                                                RestoreContext();
                                                        }
                                                }
                                        }
                                        else
                                        {
                                                if( indexTrend[bar] <
indexTrend[bar-1] )
                                                {
                                                        SetContext( idx, true );
                                                        BuyAtMarket( bar+1, "Buy " +
idx );

                                                        RestoreContext();
                                                }
                                        }
                                }
                                else
                                {
                                        if( stress[bar] <= buy )

{

                                                BuyAtMarket( bar+1, "Buy " + stock

);
                                        }
                                }
                        }
                }

                private bool SymbolIsActive(string sym)
                {
                        foreach (Position p in ActivePositions)
                                if( sym == p.Bars.Symbol )
                                        return true;
                        return false;
                }

                private int LastActivePositionInSym( List<Position> lst, string
symbol )
                {
                        return lst.FindLastIndex( delegate(Position pos) { return
pos.Symbol.Equals(symbol, StringComparison.Ordinal); });
                }
        }
}
```

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

BACK TO LIST

AMIBROKER: MARCH 2014

In “Timing The Market With Pairs Logic” in this issue, author Perry Kaufman presents a pair-trading technique based on his new stress indicator. A ready-to-use AmiBroker formula for implementing the stress indicator is presented here. To display the indicator, input the formula into the formula editor and press “apply indicator.”

A sample chart is shown in Figure 5.

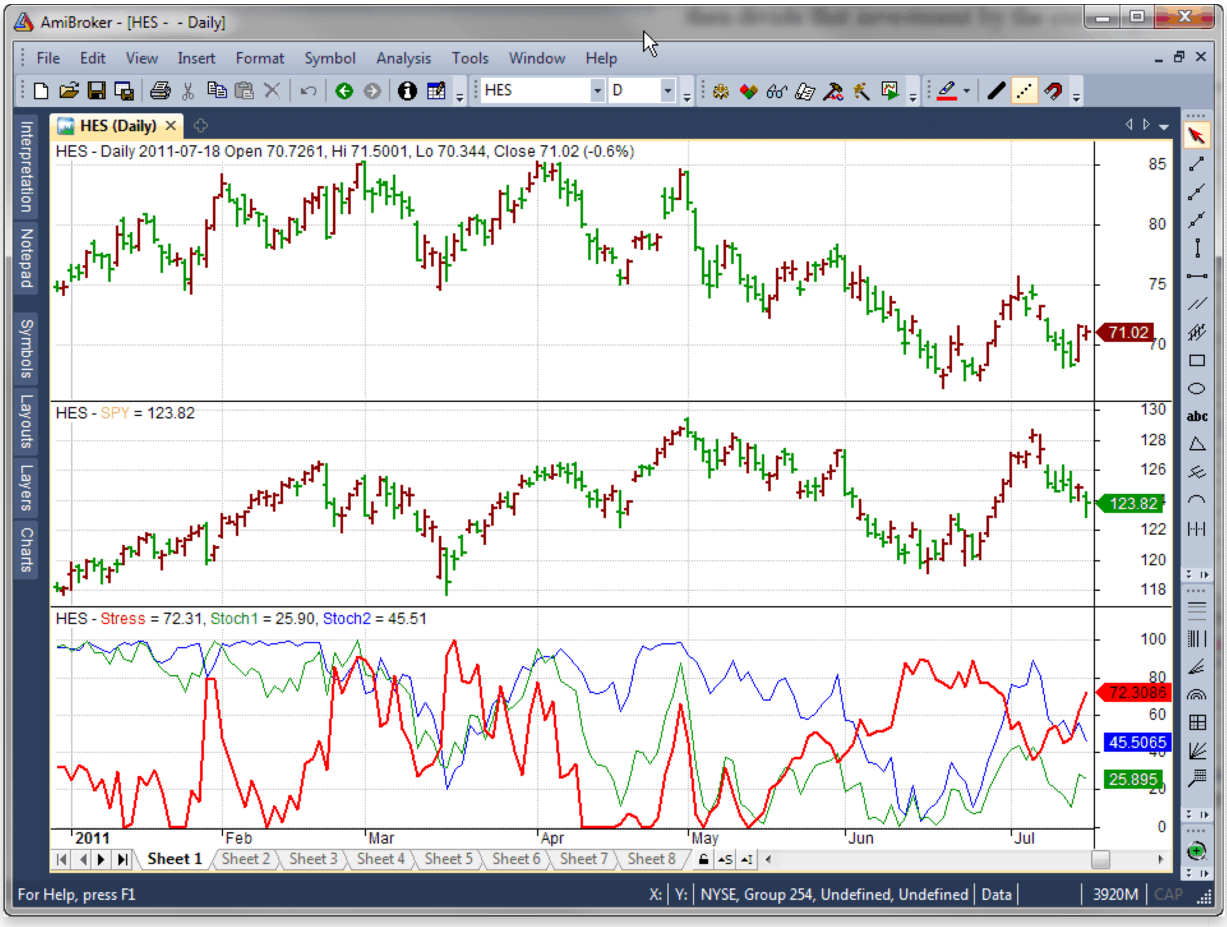


FIGURE 5: AMIBROKER. Here is a daily chart of HES with a daily chart of SPY in the middle pane and Perry Kaufman's stress indicator (red) in the bottom pane.

```
LISTING 1.
function PJKStress( High2, Low2, Close2, period )
{
  range1 = HHV( High, period ) - LLV( Low, period );
  range2 = HHV( High2, period ) - LLV( Low2, period );
  stoch1 = ( Close - LLV( Low, period ) )/range1;
  stoch2 = ( Close2 - LLV( Low2, period ) )/range2;

  VarSet("sstoch1", 100 * stoch1 );
  VarSet("sstoch2", 100 * stoch2 );

  diff = stoch1 - stoch2;

  range1 = HHV( diff, period ) - LLV( diff, period );

  return 100 * ( diff - LLV( diff, period ) )/range1;
}

momper = 60;
Hedgeper = 60;

SetForeign("SPY");
// store index data in High2/Low2/Close2 variables
```

```
High2 = High;
Low2 = Low;
Close2 = Close;
RestorePriceArrays();

stress = PJKStress( High2, Low2, Close2, momper );

Plot( stress, "Stress", colorRed, styleThick );
Plot( sstoch1, "Stoch1", colorGreen );
Plot( sstoch2, "Stoch2", colorBlue );
```

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

BACK TO LIST

NEUROSHELL TRADER: MARCH 2014

The stress indicator described by Perry Kaufman in his article in this issue (“Timing The Market With Pairs Logic”) 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 indicator:

```
Stress indicator:
SimpleStoch%K( Sub( Stoch%K(High,Low,Close,60),Stoch%K(SPDRS High,SPDRS Low,SPDRS
Close,60) ),60 )
```

To implement the stock-trading side of the pair, simply select new trading strategy from the Insert menu and enter the following in the appropriate locations of the trading strategy wizard:

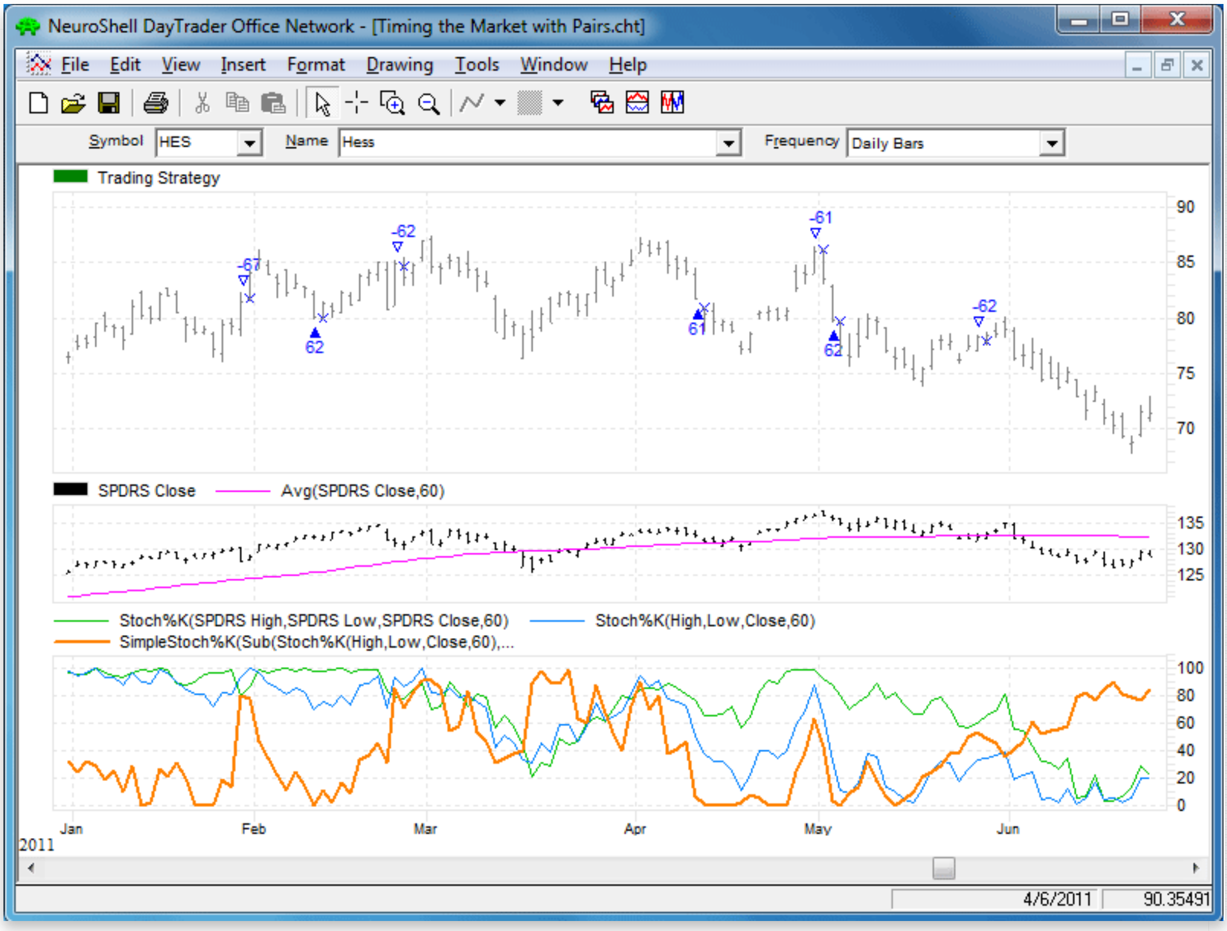
```
BUY LONG CONDITIONS: [All of which must be true]
     A<B(Stress Indicator,10)

LONG TRAILING STOP PRICES:
     TrailPrice%(Trading Strategy,20)

SELL LONG CONDITIONS: [All of which must be true]
     A>B(Stress Indicator,50)

POSITION SIZING METHOD: Fixed Dollar: 5,000.00 Dollars
```

To implement the hedge signal and calculate the size of the hedge, simply select new indicator from the Insert menu and use the indicator wizard to create the following indicators:

```
Hedge Signal:   And2( A<B( Avg(SPDRS Close,60), Lag( Avg(SPDRS Close,60), 1)), Or2(
A>B( Long Entry Signal: Trading Strategy, 0), A>B( Position(Trading Strategy,0), 0)))

Hedge Size:     Mul3( Divide(5000,Close), 0.5, Divide( StndDev( Sub( ROC(Close,1),
1), 60), StndDev( Sub( ROC(SPDRS Close,1), 1), 60)))
```

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.



NeuroShell DayTrader Office Network - [Timing the Market with Pairs.cht]

Symbol HES

Name Hess

Frequency Daily Bars

4/6/2011 90.35491

File Edit View Insert Format Drawing Tools Window Help
Trading Strategy
-67
62
-62
61
-61
62
-62
90
85
80
75
70
SPDRS Close
Avg(SPDRS Close,60)
135
130
125
Stoch%K(SPDRS High,SPDRS Low,SPDRS Close,60)
Stoch%K(High,Low,Close,60)
SimpleStoch%K(Sub(Stoch%K(High,Low,Close,60),...
100
80
60
40
20
0
Jan
Feb
Mar
Apr
May
Jun
2011

FIGURE 6: NEUROSHELL TRADER. This NeuroShell Trader chart displays the stress Indicator and corresponding stock trades.

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

BACK TO LIST

AIQ: MARCH 2014

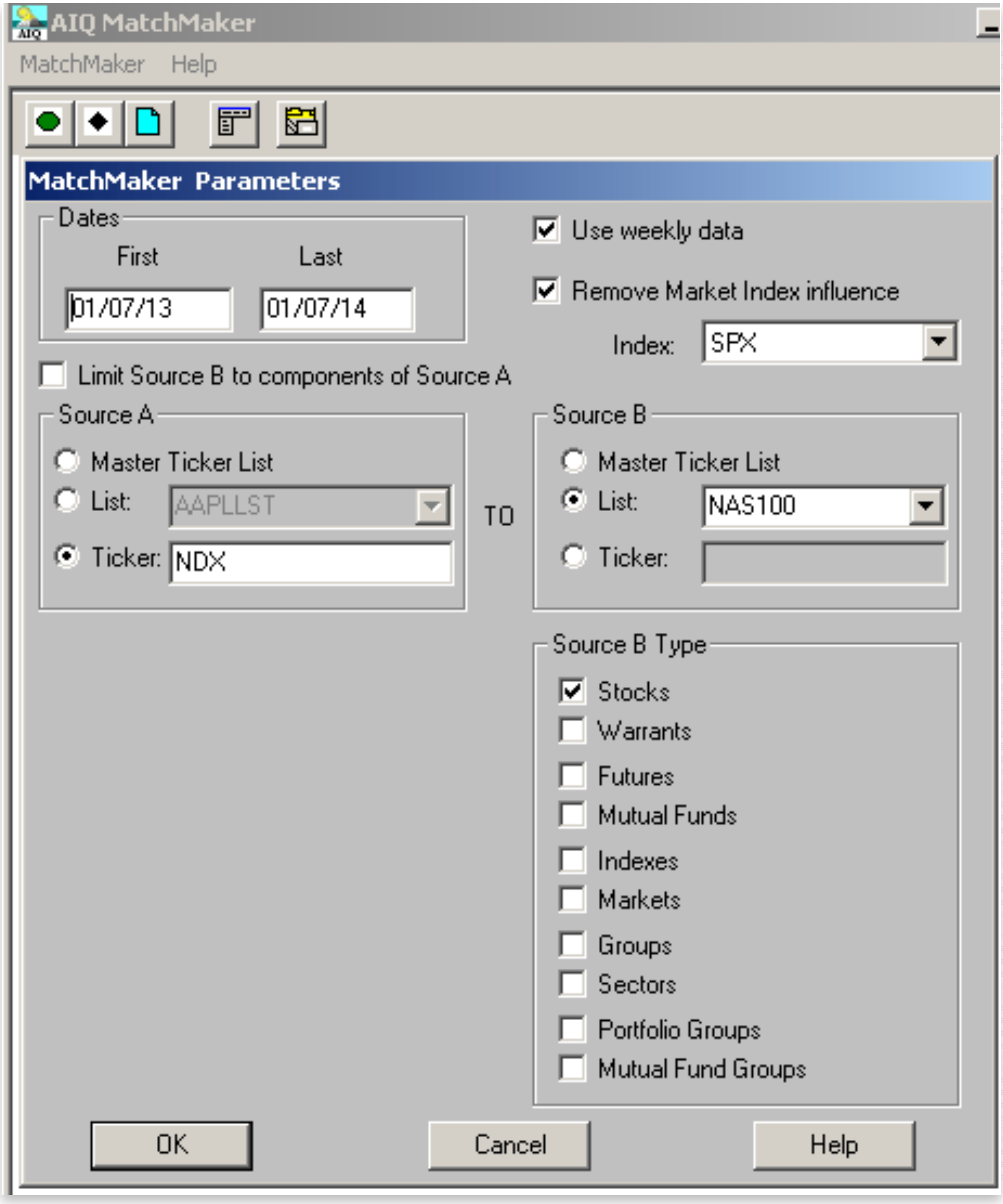
The AIQ code based on Perry Kaufman's article in this issue, "Timing The Market With Pairs Logic," is provided at www.TradersEdgeSystems.com/traderstips.htm.

The code I am providing will backtest only the long trading and will not test the hedging portion of the system. For live trading, I provided a manual input for the total value of the open positions, which would have to be computed separately and then entered daily as an input before the daily report is run once the hedge rule becomes true.

```
!TIMING THE MARKET WITH PAIRS
!Author: Perry Kaufman, TASC March 2014
!Coded by: Richard Denning 1/7/2014
!www.TradersEdgeSystems.com

!INPUTS:
stoLen is 60.
idx is "NDX".
hedgeETF is "PSQ".
eLvl is 10.
xLvl is 50.
sLoss is 0.10.
hedgeLen is 60.
hedgeRatio is 0.50.
minP is 3.
invest is 5000.
totValOpenPos is 100000. !open position value must be set manually

H is [high].
L is [low].
C is [close].
O is [open].
idxH is tickerUDF(idx,H).
idxL is tickerUDF(idx,L).
idxC is tickerUDF(idx,C).
idxO is tickerUDF(idx,O).
hedgeC is tickerUDF(hedgeETF,C).
PEP is {position entry price}.

!STRESS CODE:
rngStk is highresult(H,stoLen) - lowresult(L,stoLen).
rngIdx is highresult(idxH,stoLen) - lowresult(idxL,stoLen).
stoStk is (C - lowresult(L,stoLen)) / rngStk.
stoIdx is (idxC - lowresult(idxL,stoLen)) / rngIdx.
stoDiff is stoStk - StoIdx.
rngDiff is highresult(stoDiff,stoLen) - lowresult(stoDiff,stoLen).
stressVal is ((stoDiff - lowresult(stoDiff,stoLen)) / rngDiff) * 100.

!PAIRS SYSTEM CODE:
```

```
Buy if C > minP and countof(C > minP,4,1)=4 and stressVal <= eLvl and symbol()
<>hedgeETF.
StressExit if stressVal >= xLvl.
ExitBuy if countof(C > minP,5)<>5 or C/PEP-1 < -sLoss or StressExit.

!TREND CODE:
idxTrnd is simpleavg(idxC,hedgeLen).
Hedge if idxTrnd < valresult(idxTrnd,1) and symbol()=hedgeETF .

!SIZING CODE:
chgStk is C/valresult(C,1) - 1.
chgIdx is idxC/valresult(idxC,1) - 1.
idxSMA is simpleavg(idxC,hedgeLen).
size is floor(invest / C).
hedgeSize is floor(totValOpenPos / hedgeC*hedgeRatio) .

ShowAllValues if countof(C > minP,5)=5.
```

To get a correlated list of stocks that show good correlation to the index of choice (I used the NDX), AIQ has a *matchmaker* module that will quickly generate a list of stocks that show significant correlation to an index. In Figure 7, I show the matchmaker setup I used to quickly get a list of stocks in the NASDAQ 100 that were highly correlated to the NDX. In Figure 8, I show the results (part of which are hidden). After highlighting the ones desired for a list, simply click on the “data manager” button and a list is created, which is then used to run the tests.



Dates

First: 01/07/13

Last: 01/07/14

☑ Use weekly data

☑ Remove Market Index influence

Index: SPX

□ Limit Source B to components of Source A

Source A

○ Master Ticker List

○ List: AAPLLST

● Ticker: NDX

TO

Source B

○ Master Ticker List

● List: NAS100

○ Ticker:

Source B Type

☑ Stocks

□ Warrants

□ Futures

□ Mutual Funds

□ Indexes

□ Markets

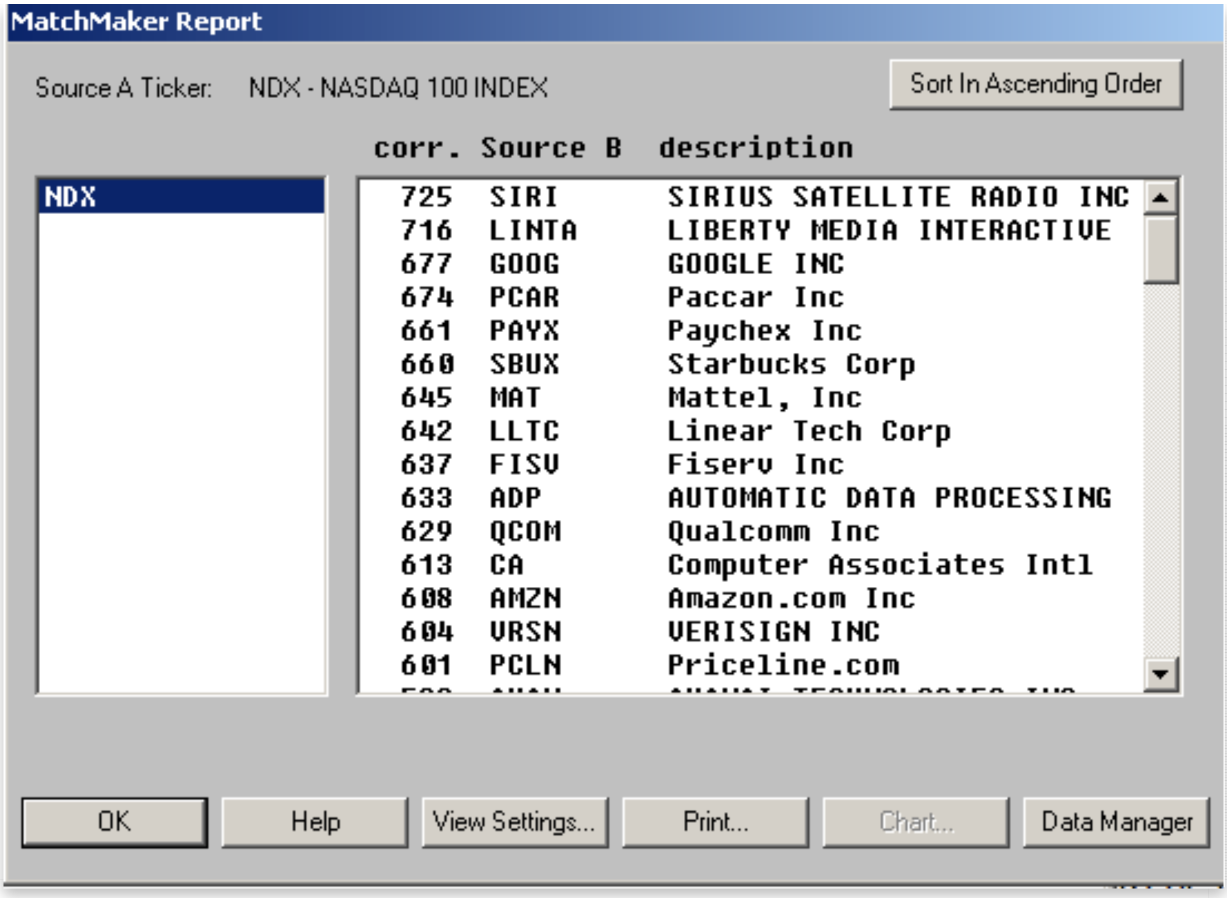
□ Groups

□ Sectors

□ Portfolio Groups

□ Mutual Fund Groups

FIGURE 7: AIQ, MATCHMAKER SETUP. Here is the setup used to get a list of stocks in the NASDAQ 100 that are highly correlated to the NDX.



Source A Ticker: NDX - NASDAQ 100 INDEX

NDX

```
corr. Source B  description
725   SIRI      SIRIUS SATELLITE RADIO INC
716   LINTA     LIBERTY MEDIA INTERACTIVE
677   GOOG      GOOGLE INC
674   PCAR      Paccar Inc
661   PAYX      Paychex Inc
660   SBUX      Starbucks Corp
645   MAT       Mattel, Inc
642   LLTC      Linear Tech Corp
637   FISV      Fiserv Inc
633   ADP       AUTOMATIC DATA PROCESSING
629   QCOM      Qualcomm Inc
613   CA        Computer Associates Intl
608   AMZN      Amazon.com Inc
604   VRSN      VERISIGN INC
601   PCLN      Priceline.com
```

FIGURE 8: AIQ, RESULTING LIST. Here are sample results of running the setup shown in Figure 7.

—Richard Denning
info@TradersEdgeSystems.com
for AIQ Systems

BACK TO LIST

TRADERSSTUDIO: MARCH 2014

The TradersStudio code based on Perry Kaufman's article in this issue, "Timing The Market With Pairs Logic," is provided at the following websites:

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

The following code file are provided in the download:

- Function PK_STRESS — returns the Kaufman stress value
- Function COUNTOF — returns the number of times a rule is true in a set lookback period
- System PK_PAIRS — system to go long stocks based on the stress indicator
- System PK_STRES_HEDGE — system that is to be used with the PK_PAIRS system for hedging
- TradePlan EqualDollar_HedgeTASC — tradeplan that runs the two systems with equal dollars invested per trade.

I set up the code on the NASDAQ 100 list of stocks and used the NDX index for pairing. I also set up the hedge using the QQQ ETF going short on the hedge signals. If trading an IRA account, the hedging system can be switched to use an inverse ETF. I used the QQQ for testing because it has more data than the inverse ETFs. In Figure 9, I show the log equity curve and the underwater percent drawdown curve over the test period 1/1/2000 to 1/8/2014. Until 2011, the max drawdowns were in the 14% area but in 2011 the max drawdown incurred was 22.7%. The compound annual return over the test period was 14.9%.

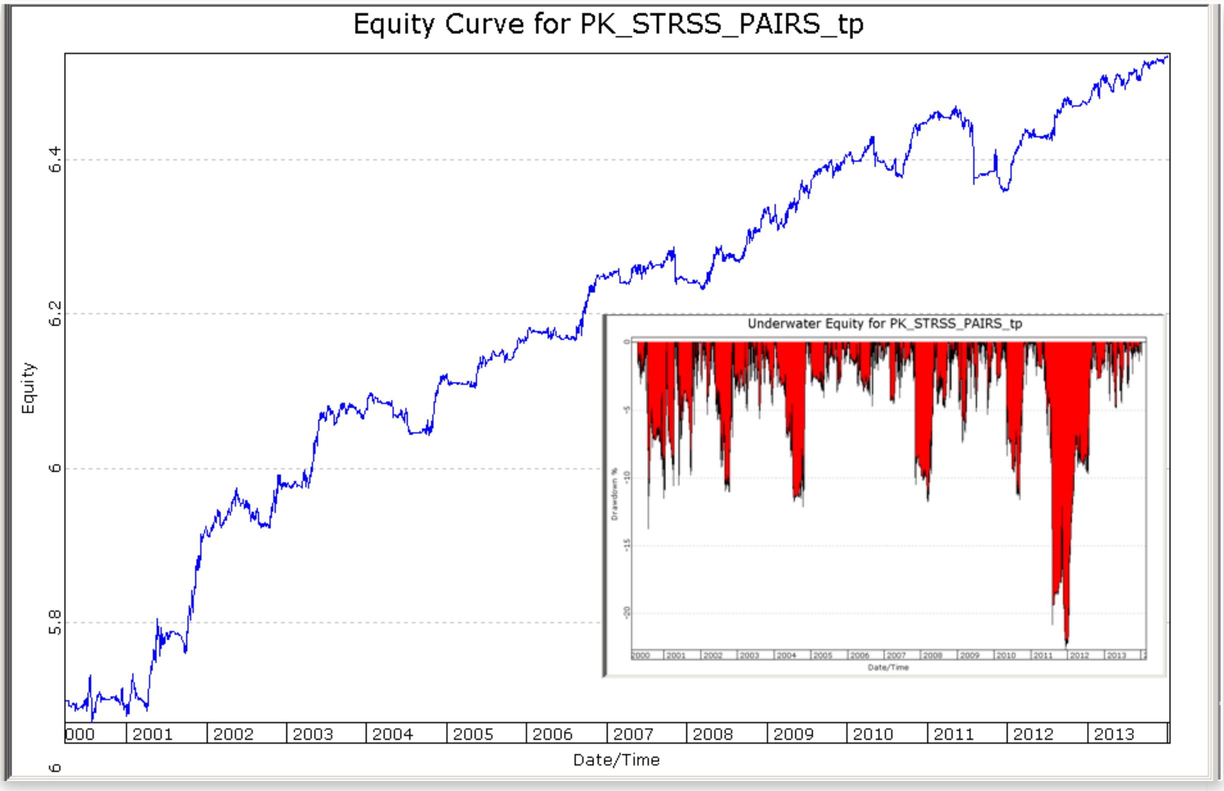


FIGURE 9: TRADERSSTUDIO. This shows the log equity curve and the underwater percent drawdown curve over the test period of 1/1/2000 to 1/8/2014 using the NASDAQ 100 list of stocks, the NDX index for pairing, and the QQQ ETF going short for hedging.

Please note that the code I have provided differs from the author's code in that the tradeplan compounds the results, so that the size is adjusted upward as the equity grows, and the hedge does not use the volatility adjustment.

The code is also shown here:

```
'TIMING THE MARKET WITH PAIRS
'Author: Perry Kaufman, TASC March 2014
'Coded by: Richard Denning 1/7/2014
'www.TradersEdgeSystems.com

Function PK_STRESS(stoLen)
Dim idxH As BarArray
Dim idxL As BarArray
Dim idxC As BarArray
Dim rngStk As BarArray
Dim rngIdx As BarArray
Dim stoStk As BarArray
Dim stoIdx As BarArray
Dim stoDiff As BarArray
Dim rngDiff As BarArray
Dim stressVal As BarArray
idxH = H Of independent1
idxL = L Of independent1
idxC = C Of independent1
rngStk = Highest(H,stoLen) - Lowest(L,stoLen)
rngIdx = Highest(idxH,stoLen) - Lowest(idxL,stoLen)
If rngStk <> 0 Then
  stoStk = (C - Lowest(L,stoLen)) / rngStk
End If
If rngIdx <> 0 Then
  stoIdx = (idxC - Lowest(idxL,stoLen)) / rngIdx
End If
stoDiff = stoStk - stoIdx
rngDiff = Highest(stoDiff,stoLen) - Lowest(stoDiff,stoLen)
If rngDiff <> 0 Then
  stressVal = ((stoDiff - Lowest(stoDiff,stoLen)) / rngDiff) * 100
End If
PK_STRESS = stressVal
End Function
'------------------------------------------------------------------
'COUNTOF Function
'returns how many times a rule is true in the lookback length
'coded by Richard Denning 01/04/08

Function COUNTOF(rule As BarArray, countLen As Integer, offset As Integer)
Dim count As Integer
Dim counter As Integer
    For counter = 0 + offset To countLen + offset - 1
        If rule[counter] Then
            count = count + 1
        End If
    Next
COUNTOF = count
End Function
'--------------------------------------------------------------------
'System to go long stocks:

Sub PK_PAIRS(stoLen,eLvl,xLvl,sLoss,hedgeLen,hedgeRatio,minP)
Dim stressVal
stressVal = PK_STRESS(stoLen)
If countof(TSCLose > minP,5,0)=5 And stressVal <= eLvl Then Buy("LE",1,0,Market,Day)
If countof(TSCLose > minP,5,0)<>5 Then ExitLong("LX_minP","",1,0,Market,Day)
If EntryPrice <> 0 Then
    If C/EntryPrice-1 < -sLoss Then ExitLong("LX_sLoss","",1,0,Market,Day)
End If
If stressVal >= xLvl Then ExitLong("LX_stress","",1,0,Market,Day)
End Sub
'--------------------------------------------------------------------
'System to hedge the long stock trading system:

Sub PK_STRESS_HEDGE(hedgeLen,offset,useInverseETF)
Dim idxTnd As BarArray
idxTnd = Average(C,hedgeLen)
If useInverseETF = 1 Then
  If idxTnd > idxTnd[offset] Then Buy("LE_hedge",1,0,Market,Day)
  If idxTnd < idxTnd[offset] Then ExitLong("LX_hedge","LE_hedge",1,0,Market,Day)
Else
  If idxTnd < idxTnd[offset] Then Sell("SE_hedge",1,0,Market,Day)
  If idxTnd > idxTnd[offset] Then ExitShort("SX_hedge","SE_hedge",1,0,Market,Day)
End If
End Sub
'--------------------------------------------------------------------
'TradePlan that runs the two sessions for the ong stock system and the hedge system:

'This will invest the same amount of money into each stock
'It will work for traderstudio stocks only
'The hedging strategy differs from thatin the article

Sub EqualDollar_HedgeTASC(leverageMult,hedgeLen,hedgeRatio)
    Dim iS As Integer
    Dim iM As Integer
    Dim iO As Integer
    Dim iT As Integer
    Dim openPos As BarArray(1000)
    Dim objS As TSProcessor.ISession
    Dim objSH As TSProcessor.ISession
    Dim objM As TSProcessor.IMarket
    Dim AO As TSProcessor.IActiveOrder
    Dim DollarPerTrade
    Dim StartAccount
    Dim DollarsPerTrade
    Dim stkChg As BarArray
    Dim idxChg As BarArray
    Dim info As Array
    Dim stkVola,idxVola,idxSize,openVolaPort,openVolaPortAvg,maxHedge
    tradeplan.StockMarginMultiplier = Min(leverageMult,2)
    objS = TradePlan.Session(0)
    objSH = TradePlan.Session(1)
    DollarsPerTrade=Round(tradeplan.SummEquity/objS.MarketCount,2)*leverageMult
For iS = 0 To tradeplan.sessioncount - 1
  If tradeplan.Session(iS) = objS Then
    For iM = 0 To objS.marketcount - 1
        objM = objS.Market(iM)
        objM.EntryNumUnits=Floor((DollarsPerTrade)/Round(objM.Data(0,"TSClose",0),4))
    Next
  End If
  If tradeplan.Session(iS) = objSH Then
    For iM = 0 To objSH.marketcount - 1
        objM = objSH.Market(iM)
        maxHedge = tradeplan.SummEquity - tradeplan.AvailableEquity

objM.EntryNumUnits=Min(hedgeRatio*maxHedge,hedgeRatio*tradeplan.SummEquity)/objM.Data(
0,"Close",0)
    Next
  Else
    objM.EntryNumUnits = 0
  End If
Next
SizeForTotalExits
End Sub
```

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

NINJATRADER: MARCH 2014

We have implemented a strategy for NinjaTrader users based on “Timing The Market With Pairs Logic” by Perry Kaufman in this issue. The strategy is available for download from www.ninjatrader.com/SC/March2014SC.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 “Timing With Pairs” file.

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

A sample chart implementing the strategy is shown in Figure 10.

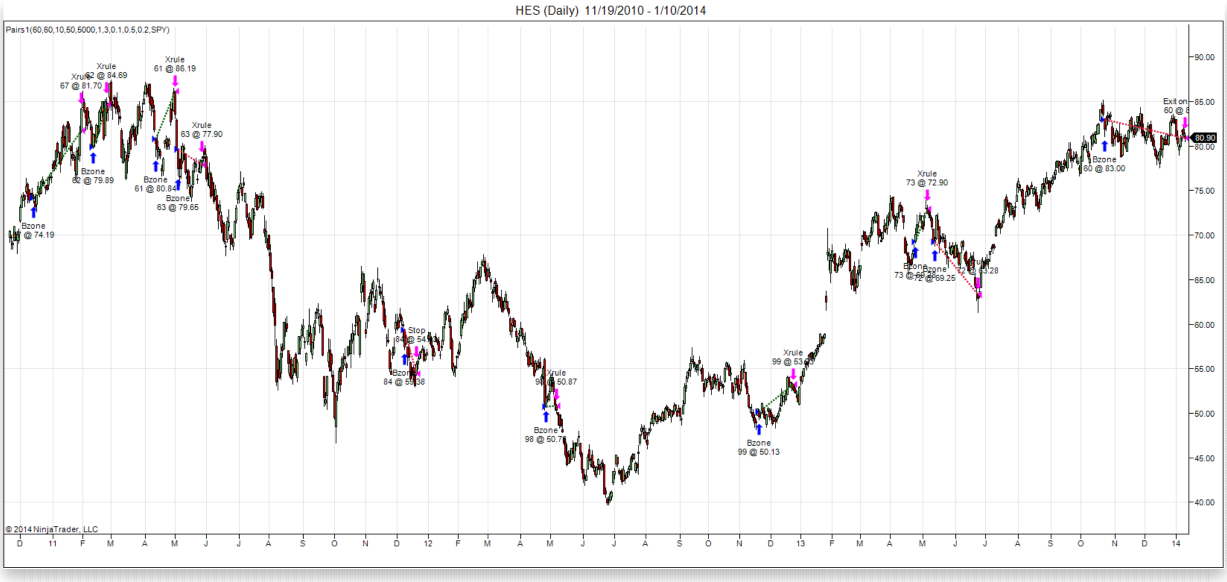


FIGURE 10: NINJATRADER. This screenshot shows the TimingWithParis strategy applied to a daily chart of the stock HES (while SPY is the second symbol internal to create the pair).

—Raymond Deux & Patrick Hodges
NinjaTrader, LLC
www.ninjatrader.com

BACK TO LIST

UPDATA: MARCH 2014

This month's Traders' Tip is based on "Timing The Market With Pairs Logic" by Perry Kaufman. In his article, Kaufman develops a correlated pair-trading system for use across fundamentally different markets to better mitigate risk across a portfolio. The key indicator in this system seeks to identify when both spread legs are at maximum divergence, and enters reversion trades at these points.

The Updata code for this 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 a firewall may paste the code shown here into the Updata custom editor and save it.

```
PLOTSTYLE LINE
INDICATORTYPE CHART
PARAMETER "Period" #PERIOD=20
PARAMETER "Stochastic 1" #STOCHPERIOD1=50
PARAMETER "Stochastic 2" #STOCHPERIOD2=50
PARAMETER "Ticker" ~TICKER=SELECT
PARAMETER "Momentum Period" #MOMPERIOD=60
PARAMETER "Entry Level" #ENTRYLEVEL=10
PARAMETER "Exit Level" #EXITLEVEL=50
PARAMETER "Min. Price" @MINPRICE=3
PARAMETER "Hedge Period" #HEDGEPERIOD=60
PARAMETER "Hedge Ratio" @HEDGERATIO=0.5
PARAMETER "Corr. Filter" @CORRFILTER=0.2
PARAMETER "Min. Price Days" #MINPRICEDAYS=5
PARAMETER "Investment" #INVESTMENT=5000
PARAMETER "Stock Cost" @COST=0
NAME "Stress" ""
@RANGE1=0
@RANGE2=0
@STOCH1=0
@STOCH2=0
@DIFF=0
@PJK_STRESS=0

@STOCKCHANGE=0
@INDEXCHANGE=0
@INDEXTREND=0
@NEWBUY=0
@STOCKSIZE=0
@INDEXSIZE=0
@INDEXPOS=0
@WAITFORRESET=0
@MINPRICEOK=0
@STOPLOSS=0
@ENTRYPRICE=0

@STOCKVOL=0
@INDEXVOL=0

@TODAYSTOCKPL=0
@TODAYINDEXPL=0
@TODAYCOMBOPL=0
@STOCKPL=0
@INDEXPL=0
@COMBOPL=0

FOR #CURDATE=0 TO #LASTDATE
   'RAW STOCHASTICS FOR BOTH LEGS OF SPREAD
   @RANGE1=PHIGH(HIGH,#PERIOD)-PLOW(LOW,#PERIOD)
   @RANGE2=PHIGH(HIGH(~TICKER,0),#PERIOD)-PLOW(LOW(~TICKER,0),#PERIOD)
   @STOCKCHANGE=-1+CLOSE/CLOSE(1)
   @INDEXCHANGE=-1+CLOSE(~TICKER,0)/CLOSE(~TICKER,1)
   @INDEXTREND=SGNL(CLOSE(~TICKER,0),#HEDGEPERIOD,M)
   @NEWBUY=FALSE
   IF @RANGE1!=0 AND @RANGE2!=0
      @STOCH1=(CLOSE-PLOW(LOW,#PERIOD))/@RANGE1
      @STOCH2=(CLOSE(~TICKER,0)-PLOW(LOW(~TICKER,0),#PERIOD))/@RANGE2
      'DIFFERENCE IN STOCHASTICS
      @DIFF=@STOCH1-@STOCH2
      'STRESS INDICATOR
      @RANGE1=PHIGH(@DIFF,#PERIOD)-PLOW(@DIFF,#PERIOD)
      IF @RANGE1!=0
         @PJK_STRESS=100*(@DIFF-PLOW(@DIFF,#PERIOD))/@RANGE1
      ENDIF
      @PLOT=@PJK_STRESS
   ENDIF
   IF ORDERISOPEN=0
      @STOCKSIZE=0
      @INDEXSIZE=0
      @INDEXPOS=0
   ENDIF
   IF @WAITFORRESET AND @PJK_STRESS>50
      @WAITFORRESET=FALSE
   ENDIF
   'SET MINIMUM PRICE CONDITION
   @MINPRICEOK=FALSE
   IF PLOW(CLOSE,#MINPRICEDAYS)>@MINPRICE
      @MINPRICEOK=TRUE
   ENDIF
   'MIN PRICE MUST OCCUR N TIMES IN A ROW TO START TO END
   IF ORDERISOPEN=0 AND @WAITFORRESET=FALSE AND @PJK_STRESS<=#ENTRYLEVEL AND
@MINPRICEOK=TRUE
      @STOCKSIZE=INT(#INVESTMENT/CLOSE)
      BUY CLOSE,@STOCKSIZE
      @NEWBUY=TRUE
      @INDEXPOS=0
      @ENTRYPRICE=CLOSE
   ELSEIF ORDERISOPEN !=0
      IF @MINPRICEOK=FALSE
         SELL CLOSE
         @INDEXPOS=0
      ELSEIF @PJK_STRESS>=#EXITLEVEL
         SELL CLOSE
         @INDEXPOS=0
      ELSEIF @STOPLOSS !=0 AND (CLOSE/@ENTRYPRICE)-1<@STOPLOSS
         SELL CLOSE
         @INDEXPOS=0
         @WAITFORRESET=TRUE
      ENDIF
   ENDIF
   'ENTER AN INDEX HEDGE IF TREND IS DOWN
   IF #HEDGEPERIOD!=0 AND @INDEXPOS=0 AND (ORDERISOPEN!=0 OR @NEWBUY!=TRUE)
      IF @INDEXTREND<HIST(@INDEXTREND,1)
         'CALC. HEDGE AMOUNT
         @STOCKVOL=STDDEV(@STOCKCHANGE,#HEDGEPERIOD)
         @INDEXVOL=STDDEV(@INDEXCHANGE,#HEDGEPERIOD)
         'IF INDEX VOL IS GREATER,RATIO LESS THAN 1
         @INDEXSIZE=@STOCKSIZE*@HEDGERATIO*(@STOCKVOL/@INDEXVOL)
         @INDEXPOS=-1
      ENDIF
   ENDIF
   'INDEX TREND TURNS UP THEN REMOVE HEDGE
   IF @INDEXPOS AND @INDEXTREND>HIST(@INDEXTREND,1)
      @INDEXPOS=0
      @INDEXSIZE=0
   ENDIF
   'THESE ITEMS ARE ALREADY HANDLES IN UPDATA
   'THE MINIMUM PRICE FILTER PREVENTS USING STOCKS THAT HAVE GONE NEGATIVE
   'DUE TO BACK TO BACK ADJUSTING SPLITS
   @TODAYSTOCKPL=ORDERISOPEN*HIST(@STOCKSIZE,1)*(CLOSE-CLOSE(1))
   IF HIST(@STOCKSIZE,1)<HIST(@STOCKSIZE,2)
      @TODAYSTOCKPL=@TODAYSTOCKPL-@COST
   ENDIF
   @TODAYINDEXPL=HIST(@INDEXSIZE,1)*HIST(@INDEXPOS,1)*(CLOSE(~TICKER,0)-CLOSE(~TICKER,1))
   IF HIST(@INDEXSIZE,1)<HIST(@INDEXSIZE,2)
      @TODAYINDEXPL=@TODAYINDEXPL-@COST
   ENDIF
   @TODAYCOMBOPL=@TODAYSTOCKPL+@TODAYINDEXPL
   @STOCKPL=@STOCKPL+@TODAYSTOCKPL
   @INDEXPL=@INDEXPL+@TODAYINDEXPL
   @COMBOPL=@COMBOPL+@TODAYCOMBOPL
NEXT
```

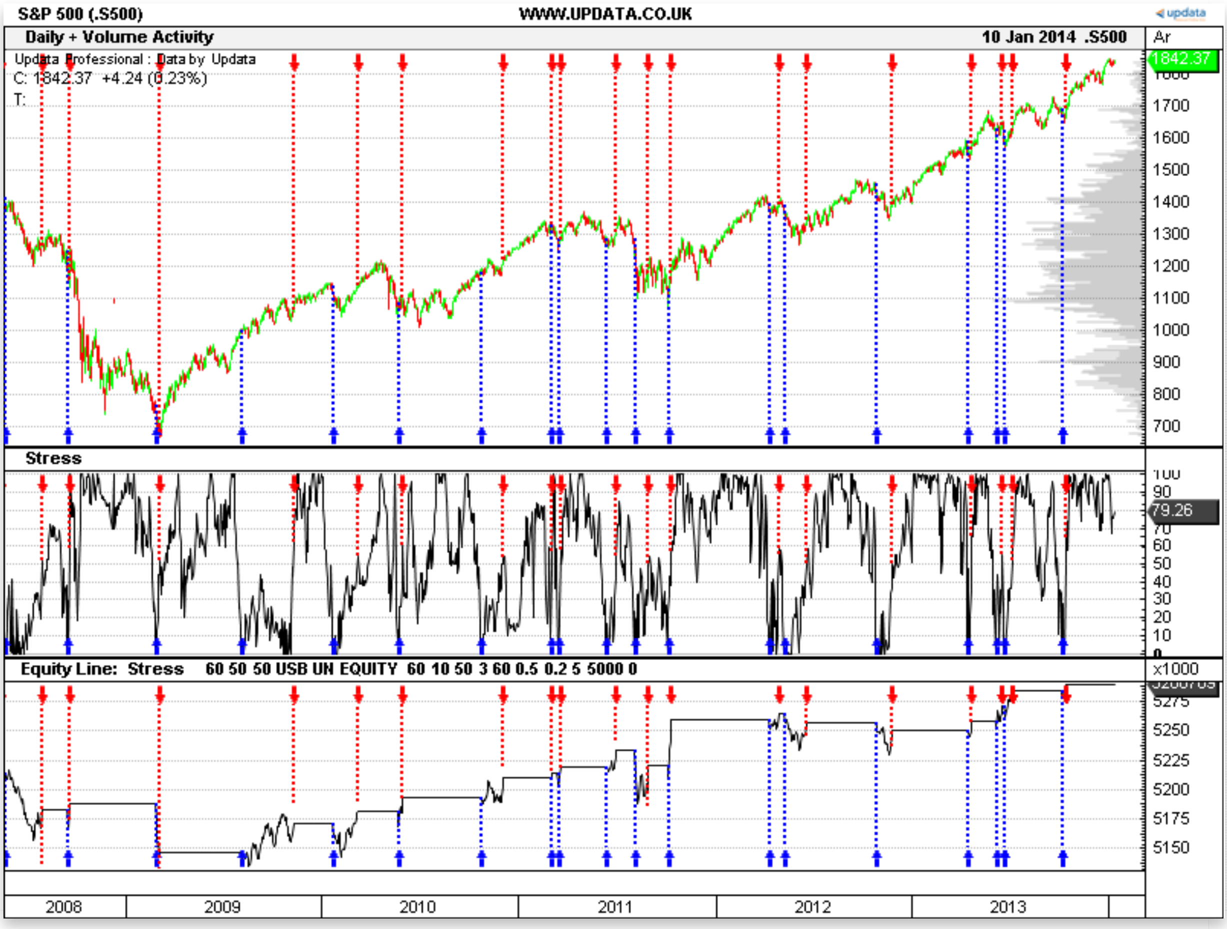


FIGURE 11: UPDATA. Here is a sample pair-trading strategy for the SPY index/USB UN equity. The stress indicator is shown in the middle pane.

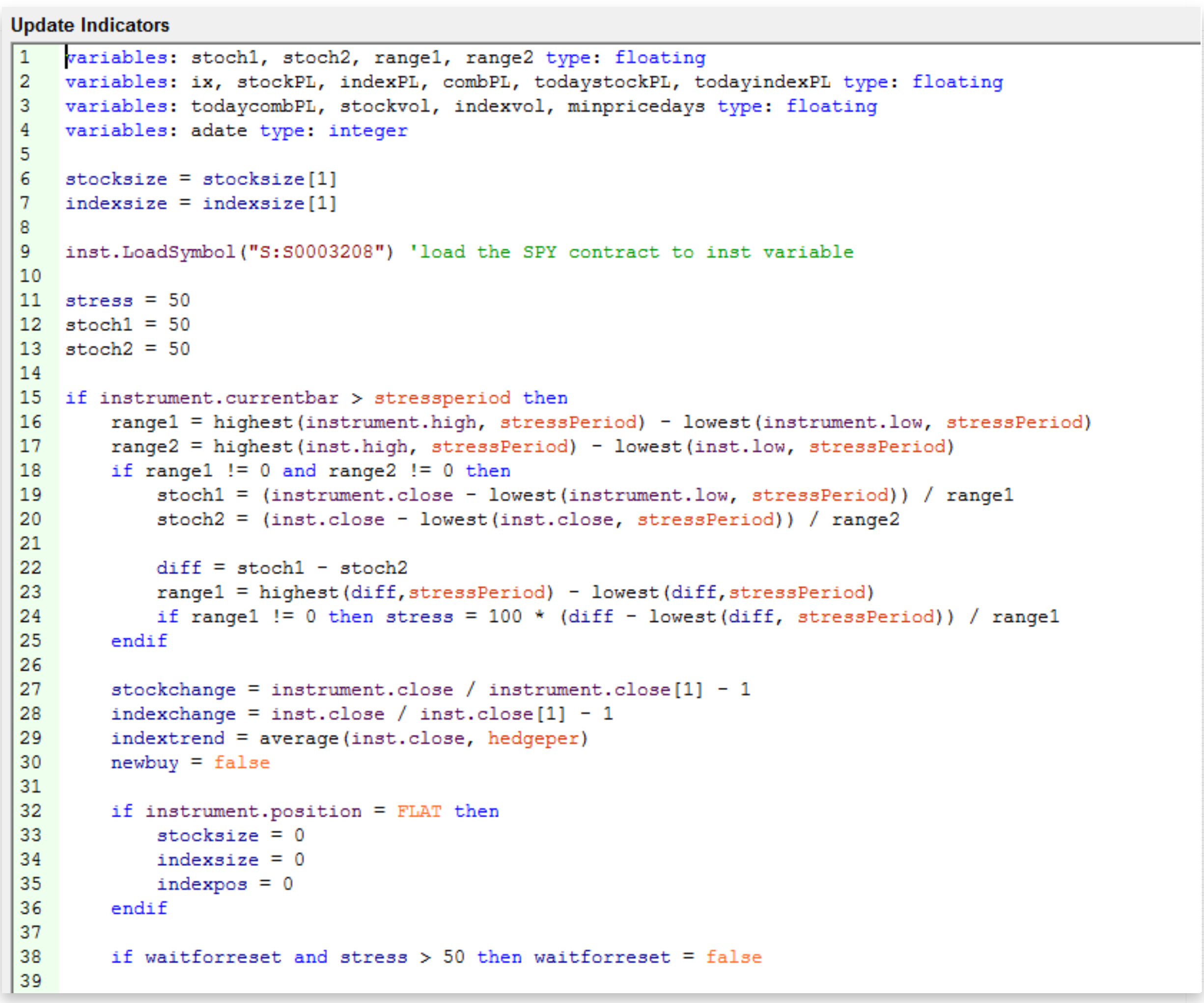
—Updata support team
support@updata.co.uk
www.updata.co.uk

BACK TO LIST

TRADING BLOX: MARCH 2014

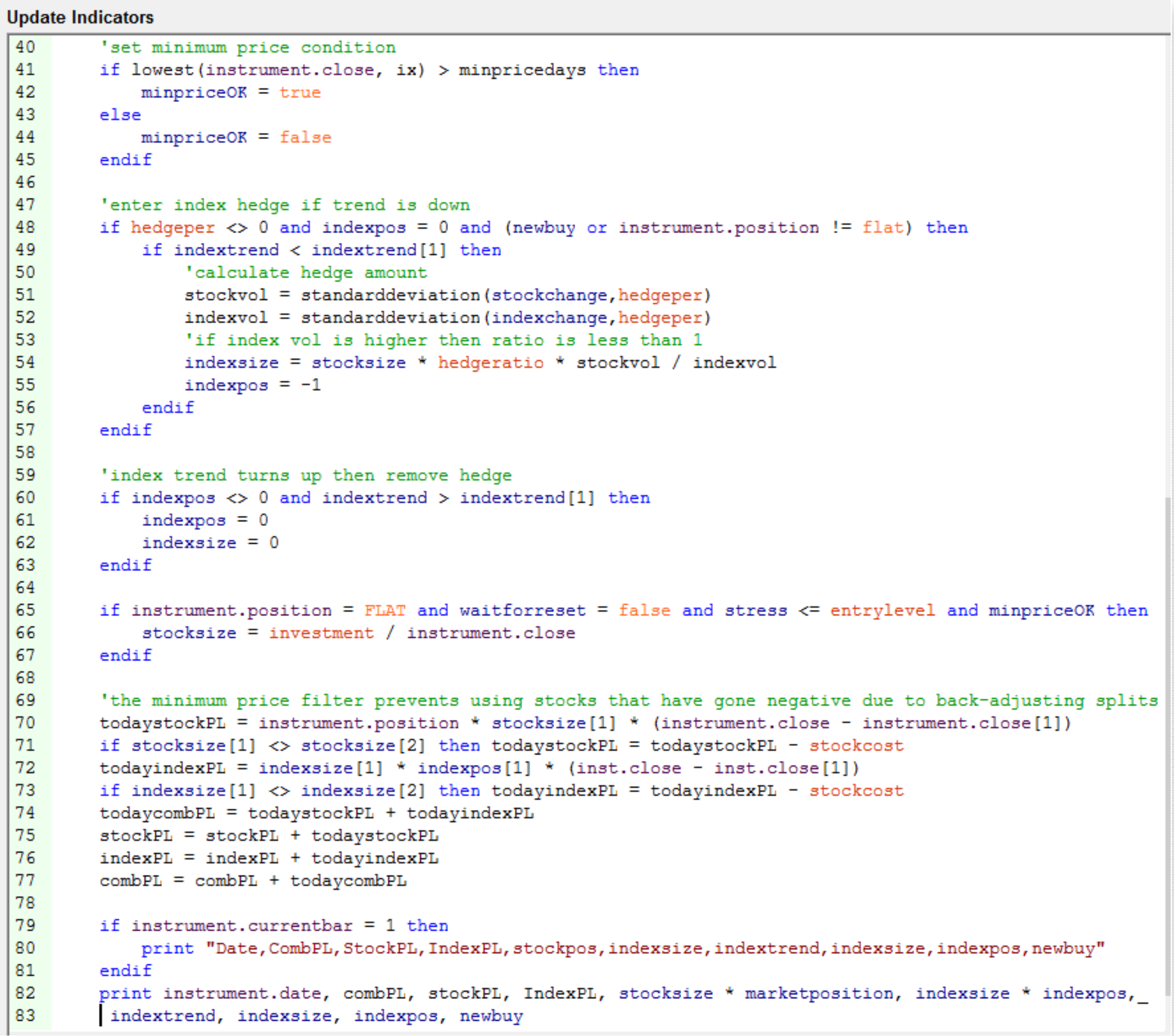
In “Timing The Market With Pairs Logic” in this issue, author Perry Kaufman presents his stress indicator and a pair-trading strategy. For the Trading Blox code that replicates these strategies, visit the Trading Blox forum at http://www.tradingblox.com/forum/viewtopic.php?t=10172.

The code is also shown here:



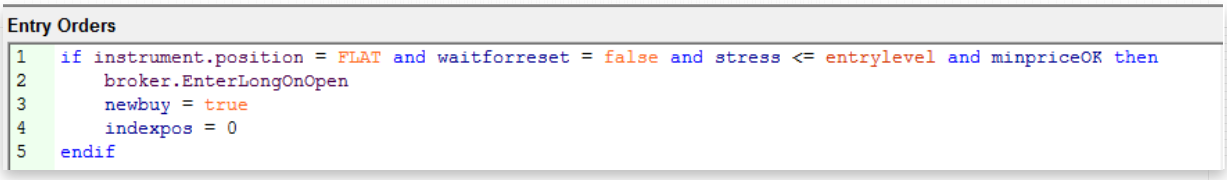
```
variables: stoch1, stoch2, range1, range2 type: floating
variables: ix, stockPL, indexPL, combPL, todaystockPL, todayindexPL type: floating
variables: todaycombPL, stockvol, indexvol, minpricedays type: floating
variables: adate type: integer

stocksize = stocksize[1]
indexsize = indexsize[1]

inst.LoadSymbol("S:S0003208") 'load the SPY contract to inst variable

stress = 50
stoch1 = 50
stoch2 = 50

if instrument.currentbar > stressperiod then
    range1 = highest(instrument.high, stressPeriod) - lowest(instrument.low, stressPeriod)
    range2 = highest(inst.high, stressPeriod) - lowest(inst.low, stressPeriod)
    if range1 != 0 and range2 != 0 then
        stoch1 = (instrument.close - lowest(instrument.low, stressPeriod)) / range1
        stoch2 = (inst.close - lowest(inst.close, stressPeriod)) / range2

        diff = stoch1 - stoch2
        range1 = highest(diff,stressPeriod) - lowest(diff,stressPeriod)
        if range1 != 0 then stress = 100 * (diff - lowest(diff, stressPeriod)) / range1
    endif

    stockchange = instrument.close / instrument.close[1] - 1
    indexchange = inst.close / inst.close[1] - 1
    indextrend = average(inst.close, hedgeper)
    newbuy = false

    if instrument.position = FLAT then
        stocksize = 0
        indexsize = 0
        indexpos = 0
    endif

    if waitforreset and stress > 50 then waitforreset = false

```

Update Indicators code part 1.



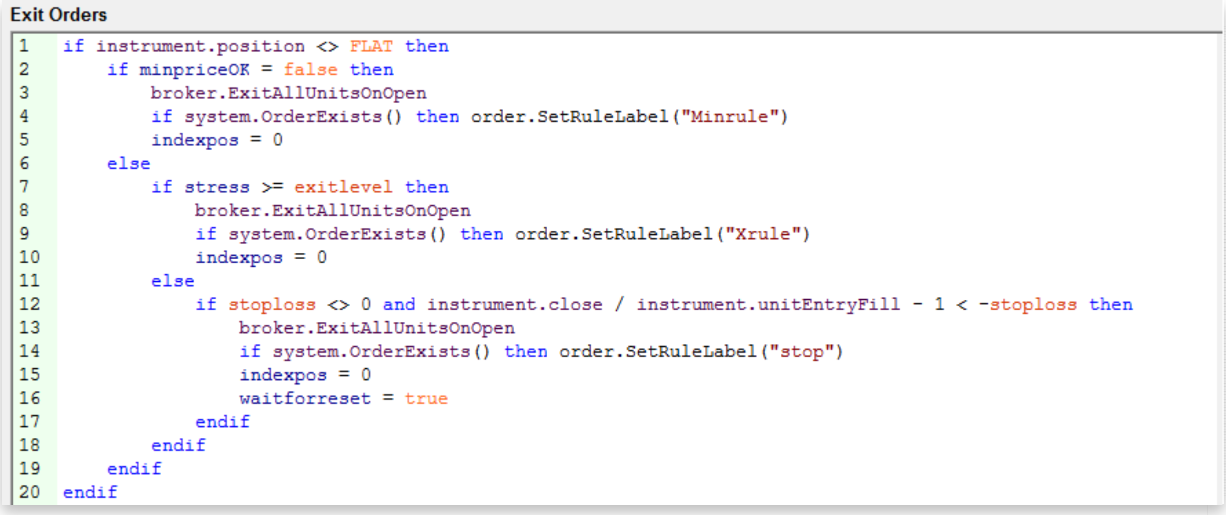
```
    'set minimum price condition
    if lowest(instrument.close, ix) > minpricedays then
        minpriceOK = true
    else
        minpriceOK = false
    endif

    'enter index hedge if trend is down
    if hedgeper <> 0 and indexpos = 0 and (newbuy or instrument.position != flat) then
        if indextrend < indextrend[1] then
            'calculate hedge amount
            stockvol = standarddeviation(stockchange,hedgeper)
            indexvol = standarddeviation(indexchange,hedgeper)
            'if index vol is higher then ratio is less than 1
            indexsize = stocksize * hedgeratio * stockvol / indexvol
            indexpos = -1
        endif
    endif

    'index trend turns up then remove hedge
    if indexpos <> 0 and indextrend > indextrend[1] then
        indexpos = 0
        indexsize = 0
    endif

    if instrument.position = FLAT and waitforreset = false and stress <= entrylevel and minpriceOK then
        stocksize = investment / instrument.close
    endif

    'the minimum price filter prevents using stocks that have gone negative due to back-adjusting splits
    todaystockPL = instrument.position * stocksize[1] * (instrument.close - instrument.close[1])
    if stocksize[1] <> stocksize[2] then todaystockPL = todaystockPL - stockcost
    todayindexPL = indexsize[1] * indexpos[1] * (inst.close - inst.close[1])
    if indexsize[1] <> indexsize[2] then todayindexPL = todayindexPL - stockcost
    todaycombPL = todaystockPL + todayindexPL
    stockPL = stockPL + todaystockPL
    indexPL = indexPL + todayindexPL
    combPL = combPL + todaycombPL

    if instrument.currentbar = 1 then
        print "Date,CombPL,StockPL,IndexPL,stockpos,indexsize,indextrend,indexsize,indexpos,newbuy"
    endif
    print instrument.date, combPL, stockPL, IndexPL, stocksize * marketposition, indexsize * indexpos,_
     indextrend, indexsize, indexpos, newbuy
```

Update Indicators code part 2.



```
if instrument.position = FLAT and waitforreset = false and stress <= entrylevel and minpriceOK then
    broker.EnterLongOnOpen
    newbuy = true
    indexpos = 0
endif
```

Entry Orders code.



Exit Orders

```
if instrument.position <> FLAT then
    if minpriceOK = false then
        broker.ExitAllUnitsOnOpen
        if system.OrderExists() then order.SetRuleLabel("Minrule")
        indexpos = 0
    else
        if stress >= exitlevel then
            broker.ExitAllUnitsOnOpen
            if system.OrderExists() then order.SetRuleLabel("Xrule")
            indexpos = 0
        else
            if stoploss <> 0 and instrument.close / instrument.unitEntryFill - 1 < -stoploss then
                broker.ExitAllUnitsOnOpen
                if system.OrderExists() then order.SetRuleLabel("stop")
                indexpos = 0
                waitforreset = true
            endif
        endif
    endif
endif
```

Exit Orders code.

—Trading Blox
tradingblox.com

BACK TO LIST

TRADE NAVIGATOR: MARCH 2014

Based on “Timing The Market With Pairs Logic” by Perry Kaufman in this issue, we have created the special file “SC201403” that Trade Navigator users can download to make it easy to implement Kaufman’s technique.

To download the file, click on Trade Navigator’s blue telephone button, select *download special file*, then erase the word “upgrade” and type in SC201403, and click the start button. When prompted to upgrade, click the yes button. If prompted to close all software, click the continue button. Your library will now download.

This library contains all the indicators discussed in Kaufman’s article as well as a template called “timing the market with pairs.” This prebuilt template can be overlaid onto a chart by opening the charting dropdown menu, then selecting the templates command, then selecting the template. This template, when applied to a stock chart, will contain the stock’s price in the upper pane, the SPY price in the central pane for comparison, and the three stress indicators in the lower pane.

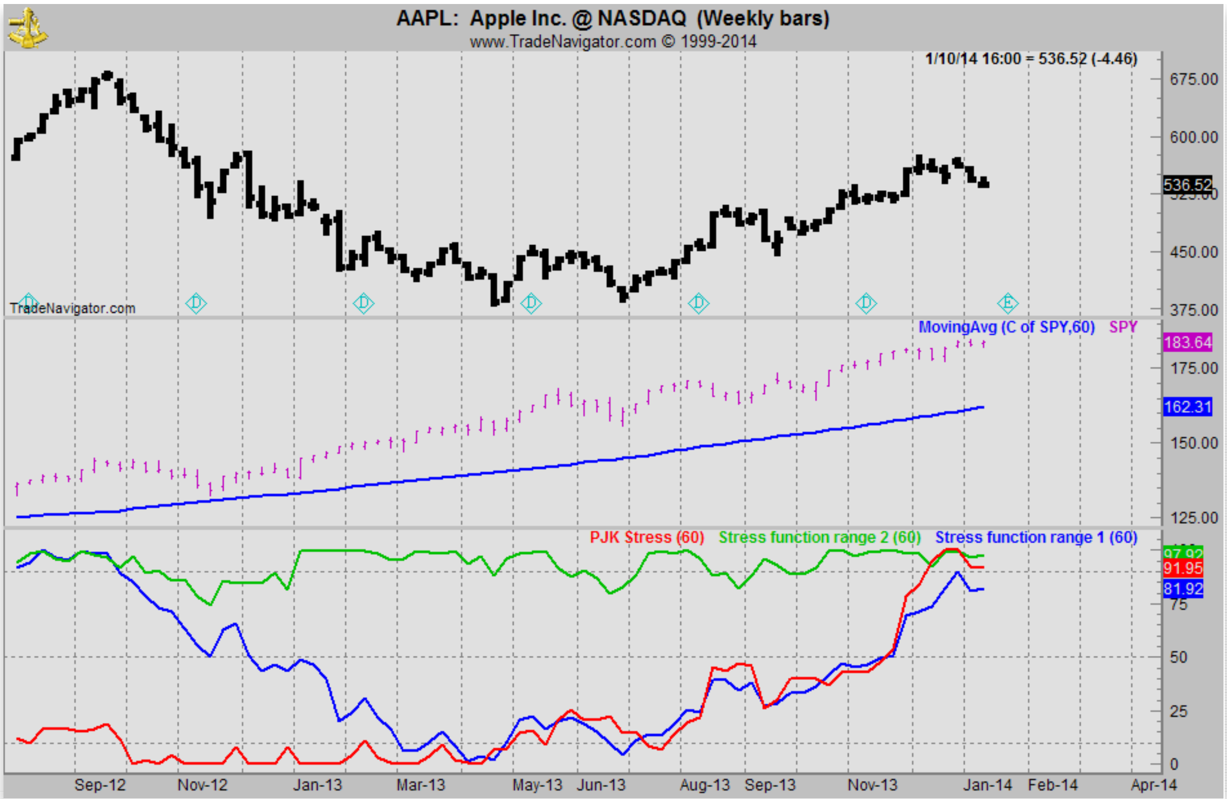


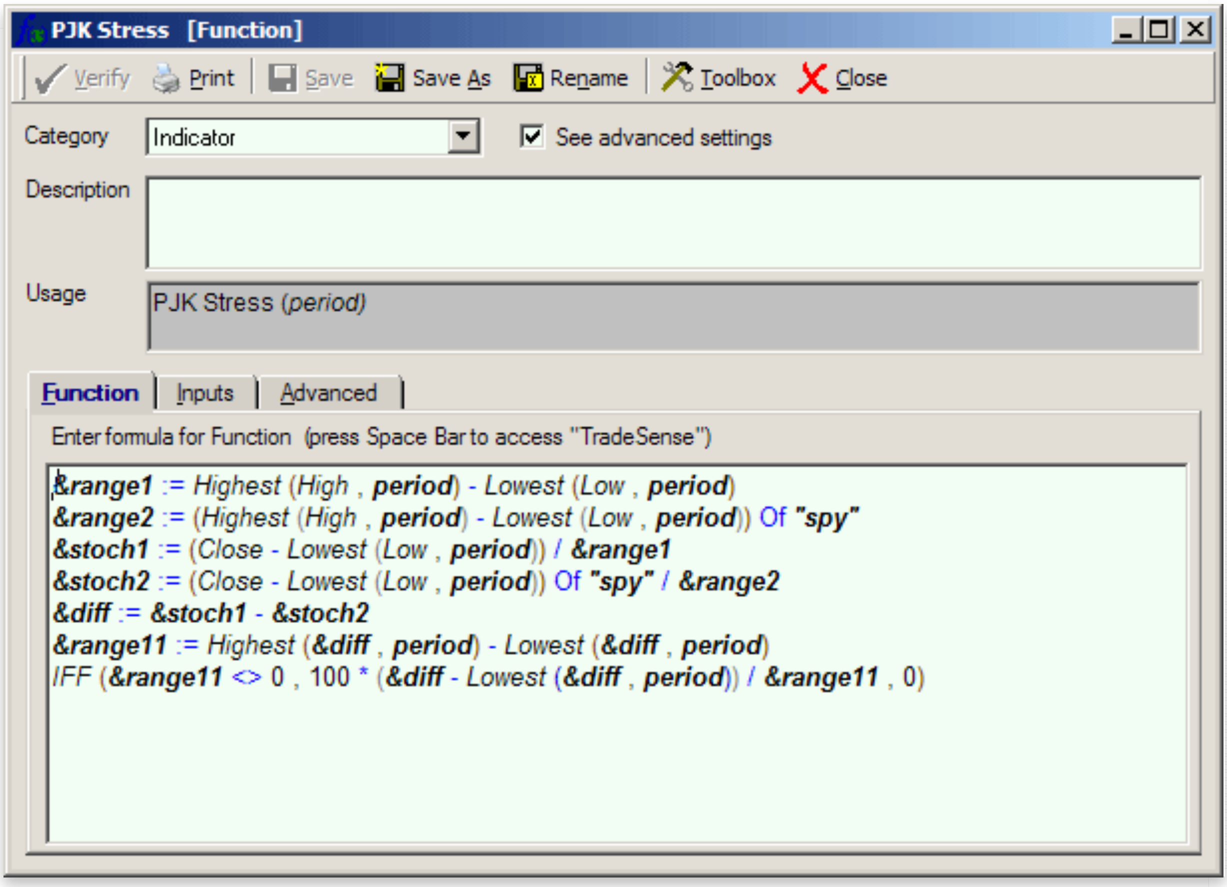
FIGURE 12: TRADE NAVIGATOR, STRESS INDICATORS. Here is a sample chart of AAPL showing the three stress functions (the PJK stress indicator, the stress function range 1, and stress function range 2).

The TradeSense code for the indicators follows:

```
PJK STRESS
&range1 := Highest (High , period) - Lowest (Low , period)
&range2 := (Highest (High , period) - Lowest (Low , period)) Of "spy"
&stoch1 := (Close - Lowest (Low , period)) / &range1
&stoch2 := (Close - Lowest (Low , period)) Of "spy" / &range2
&diff := &stoch1 - &stoch2
&range11 := Highest (&diff , period) - Lowest (&diff , period)
IFF (&range11 <> 0 , 100 * (&diff - Lowest (&diff , period)) / &range11 , 0)

STRESS FUNCTION RANGE 1
&range1 := Highest (High , period) - Lowest (Low , period)
&range2 := (Highest (High , period) - Lowest (Low , period)) Of "spy"
&stoch1 := (Close - Lowest (Low , period)) / &range1
&stoch1 * 100

STRESS FUNCTION RANGE 2
&range1 := Highest (High , period) - Lowest (Low , period)
&range2 := (Highest (High , period) - Lowest (Low , period)) Of "spy"
&stoch2 := (Close - Lowest (Low , period)) Of "spy" / &range2
&stoch2 * 100
```

CREATING A FUNCTION

To set up these indicators manually, click on the *edit* dropdown menu and open the *trader's toolbox* (or use CTRL+T) and click on the *functions* tab. Next, click *new* and a "new function" dialog window will open. In its text box, input the code for one of the above indicators. Ensure there are no extra spaces at the end of each line. When this is completed, click on the verify button. You may be presented with an *add inputs* pop-up 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 well, when you click on the function tab, the code you entered will convert to italics. Finally, click the save button and type a name for the indicator.



Category: Indicator

☑ See advanced settings

Description:

Usage: PJK Stress *(period)*

Enter formula for Function (press Space Bar to access "TradeSense")

```
&range1 := Highest (High , period) - Lowest (Low , period)
&range2 := (Highest (High , period) - Lowest (Low , period)) Of "spy"
&stoch1 := (Close - Lowest (Low , period)) / &range1
&stoch2 := (Close - Lowest (Low , period)) Of "spy" / &range2
&diff := &stoch1 - &stoch2
&range11 := Highest (&diff , period) - Lowest (&diff , period)
IFF (&range11 <> 0 , 100 * (&diff - Lowest (&diff , period)) / &range11 , 0)
```

FIGURE 13: TRADE NAVIGATOR, INDICATOR RULES.
This shows the rules for the PJK stress function.

EDITING A CHART

Once you have created the indicators, you can insert the indicators onto your chart by opening the charting dropdown menu, select the "add to chart" command, then on the *indicators* tab, find your named indicator, select it, and click on the "add" button. Repeat this procedure for the other indicators as well if you wish.

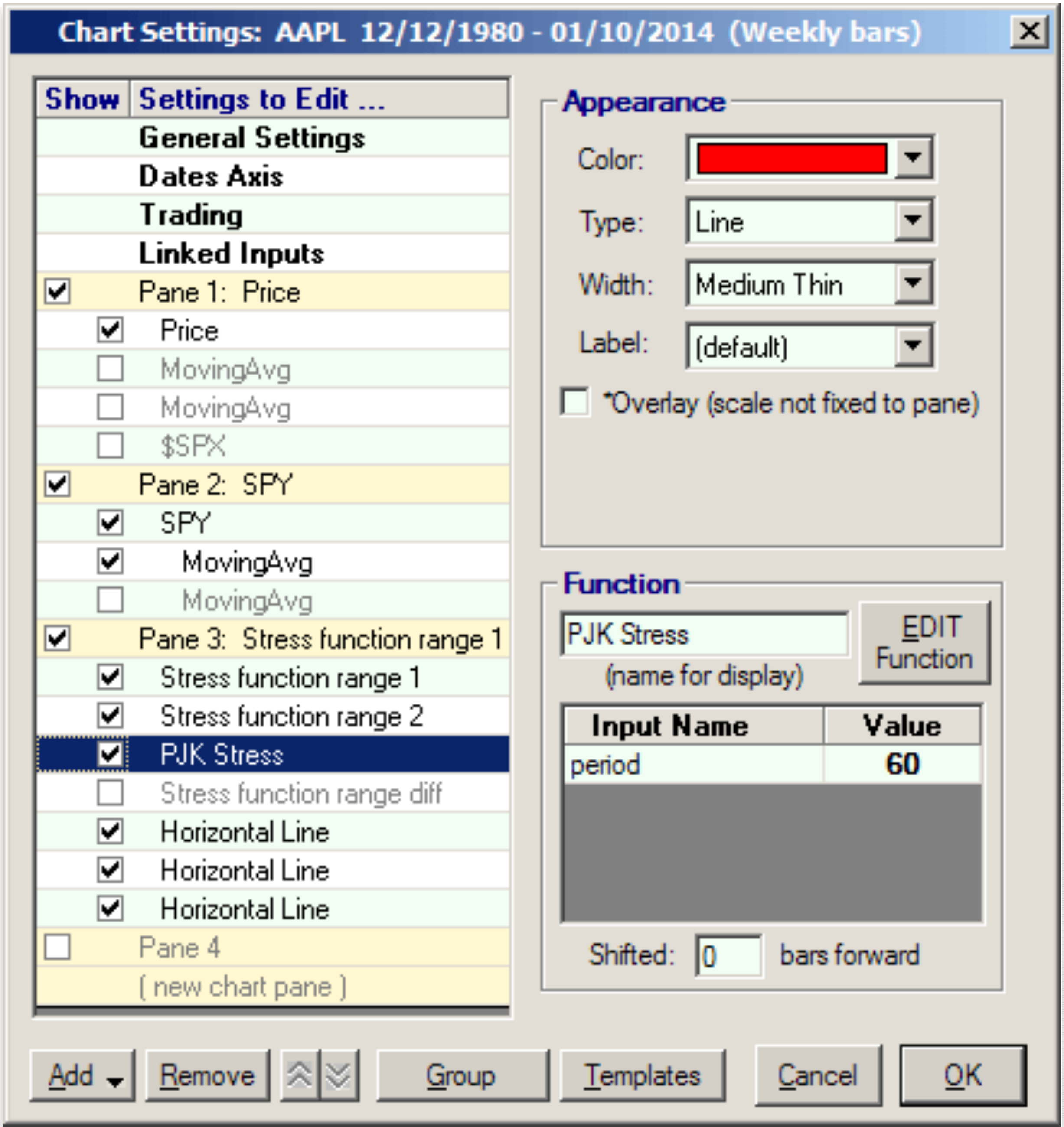


Chart Settings: AAPL 12/12/1980 - 01/10/2014 (Weekly bars)

Show	Settings to Edit ...
	General Settings
	Dates Axis
	Trading
	Linked Inputs
☑	Pane 1: Price
☑	Price
□	MovingAvg
□	MovingAvg
□	$SPX
☑	Pane 2: SPY
☑	SPY
☑	MovingAvg
□	MovingAvg
☑	Pane 3: Stress function range 1
☑	Stress function range 1
☑	Stress function range 2
☑	PJK Stress
□	Stress function range diff
☑	Horizontal Line
☑	Horizontal Line
☑	Horizontal Line
□	Pane 4
	(new chart pane)

Appearance

Color: 🟥

Type: Line

Width: Medium Thin

Label: (default)

□ *Overlay (scale not fixed to pane)

Function

PJK Stress

(name for display)

Input Name	Value
period	60

Shifted: 0 bars forward

FIGURE 14: TRADE NAVIGATOR, CHART SETTINGS.
Here is the chart settings window showing how to edit inputs.

If you need assistance with recreating the chart template settings discussed here, contact our technical support by clicking on the *live chat tool* located under Trade Navigator's help menu and also near the top of our homepage, www.TradeNavigator.com, or call our technical support at (719) 884-0245 during our business hours. Happy trading!

—Robert Giacolono
Genesis Financial Technologies
719 884-0245, www.TradeNavigator.com

BACK TO LIST

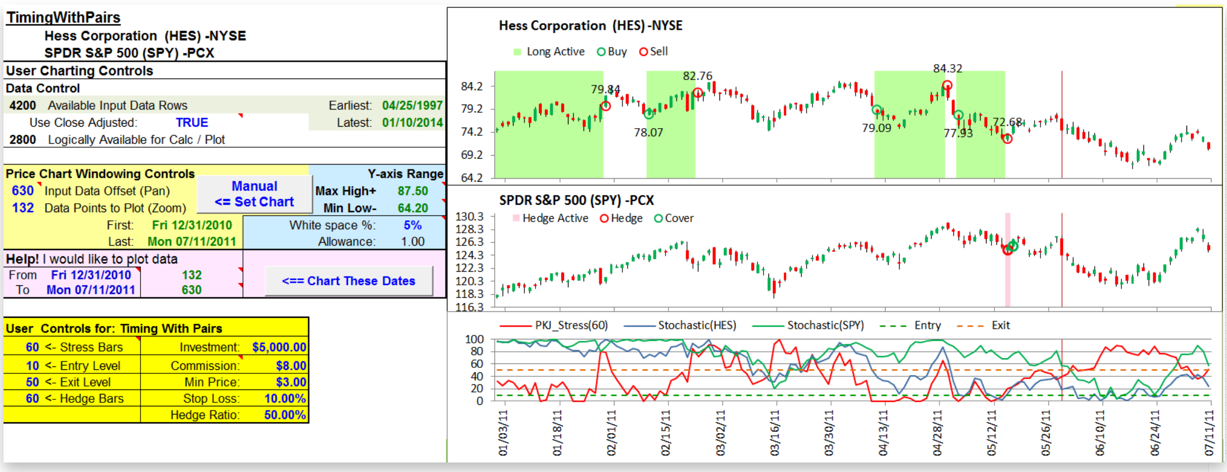
MICROSOFT EXCEL: MARCH 2014

In “Timing The Market With Pairs Logic” in this issue, author Perry Kaufman shows an accessible approach to relative value arbitrage.

His stress indicator shows us when a given equity is oversold relative to the overall market as represented by the SPY (SPDR S&P 500). This is treated as a buy opportunity.

The stress indicator also shows when the equity no longer holds an advantage relative to the market. This is one of three possible exit triggers for the equity holdings.

Finally, Kaufman shows us a method that may be used to control overall risk by scaling into trades against the index as a hedge to the equity transaction (Figure 15).



TimingWithPairs
Hess Corporation (HES) -NYSE
SPDR S&P 500 (SPY) -PCX

User Charting Controls
Data Control
4200 Available Input Data Rows — Earliest: 04/25/1997
Use Close Adjusted: TRUE — Latest: 01/10/2014
2800 Logically Available for Calc / Plot

Price Chart Windowing Controls
630 Input Data Offset (Pan)
132 Data Points to Plot (Zoom)
First: Fri 12/31/2010
Last: Mon 07/11/2011
Y-axis Range
Max High+ 87.50
Min Low- 64.20
White space %: 5%
Allowance: 1.00

Help! I would like to plot data
From Fri 12/31/2010 132
To Mon 07/11/2011 630

User Controls for: Timing With Pairs	
60 <- Stress Bars	Investment: $5,000.00
10 <- Entry Level	Commission: $8.00
50 <- Exit Level	Min Price: $3.00
60 <- Hedge Bars	Stop Loss: 10.00%
	Hedge Ratio: 50.00%

FIGURE 15: EXCEL, Profit & Loss. This view shows the stock, index, and hedged combination.

Figure 16 shows a sample chart illustrating the stress indicator plotted alongside the SPY and a sample stock (HES).

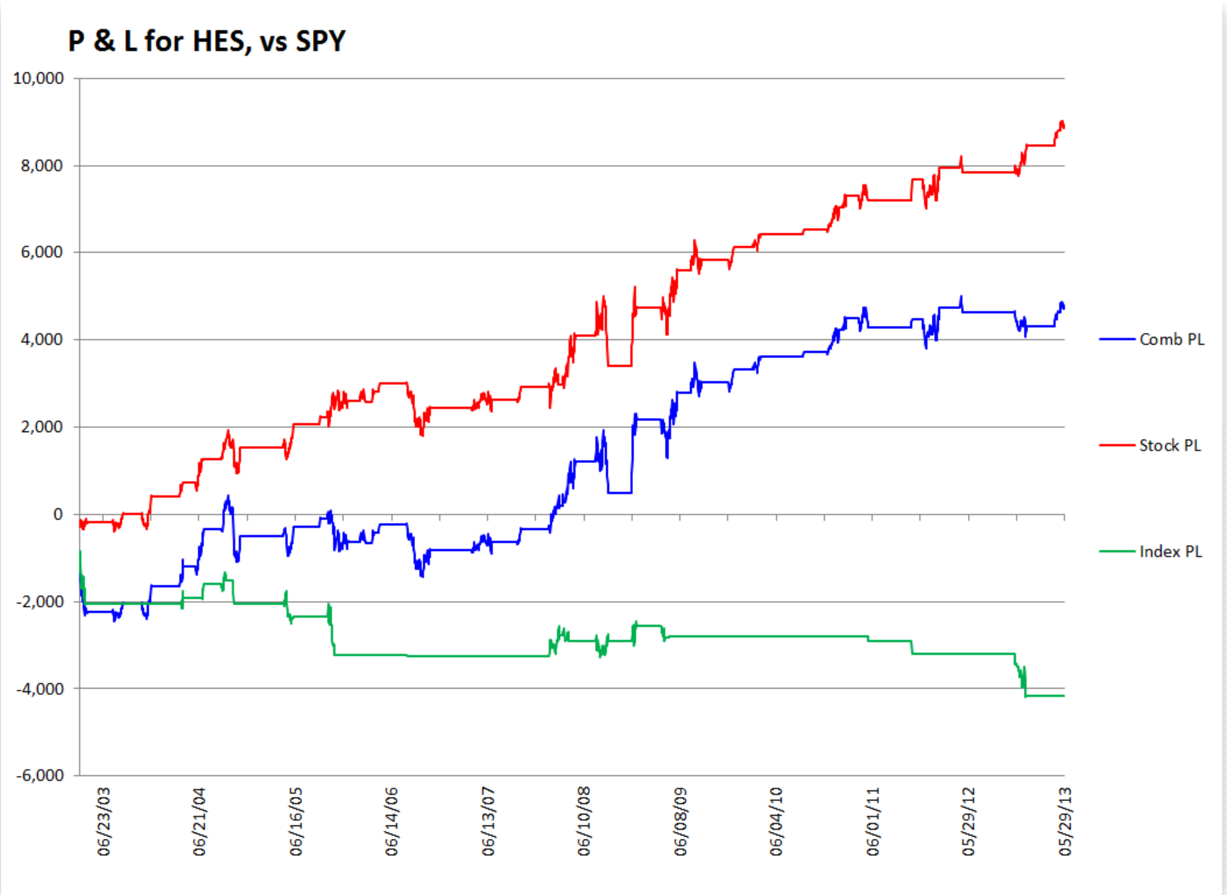


FIGURE 16: EXCEL, STRESS INDICATOR AND HES vs. spy. This chart shows the stress indicator for the pair HES vs. SPY.

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" 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 2014 issue of
Technical Analysis of STOCKS & COMMODITIES magazine.

April 2014

For this month's Traders' Tips, the focus is Melvin Dickover's article in this issue, "Evidence-Based Support & Resistance." Here we present the April 2014 Traders' Tips code with possible implementations in various software.

Code for OmniTrader is already provided in Melvin Dickover's article. Subscribers will find that code at the Subscriber Area of our website. (Click on "Article Code" from the S&C menu.) Presented here is an overview of possible implementations for other software.

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

TRADESTATION: APRIL 2014

In "Evidence-Based Support & Resistance" in this issue, author Melvin Dickover introduces two new indicators to help traders note support and resistance areas by identifying supply and demand pools. Here, we present EasyLanguage code for the RelativeVolume and Freedom of Movement (FoM) indicators for TradeStation.

RelativeVolume (Indicator)

```
{
Reference: Technical Analysis of
Stocks & Commodities, Apr 2014.
Article: Evidence-Based Support & Resistance
Indicator Name : RelativeVolume
}
```

```
inputs:
        int Period( 60 ),
        double NumStdDevs( 2 ),
        int NormalVolumeColor( LightGray ),
        int VolumeSpikeColor( Black ),
        bool AllowNegativePlots( false ) ;
variables:
        double MyVolume( 0 ),
        double AvgVolume( 0 ),
        double StdDevVol( 0 ),
        double RelVolume( 0 ) ;

if BarType >= 2 and BarType < 5 then
        MyVolume = Volume
else
        MyVolume =      Ticks ;

AvgVolume = Average( MyVolume, Period ) ;
StdDevVol = StdDev( MyVolume, Period ) ;

RelVolume = iff( StdDevVol <> 0 ,
( MyVolume - AvgVolume ) / StdDevVol, 0 ) ;

if AllowNegativePlots = false then
        RelVolume = MaxList( RelVolume, 0 ) ;

Plot1( RelVolume, "RelVolume" ) ;

if RelVolume > NumStdDevs then
        SetPlotColor( 1, VolumeSpikeColor )
else
        SetPlotColor( 1, NormalVolumeColor ) ;
```

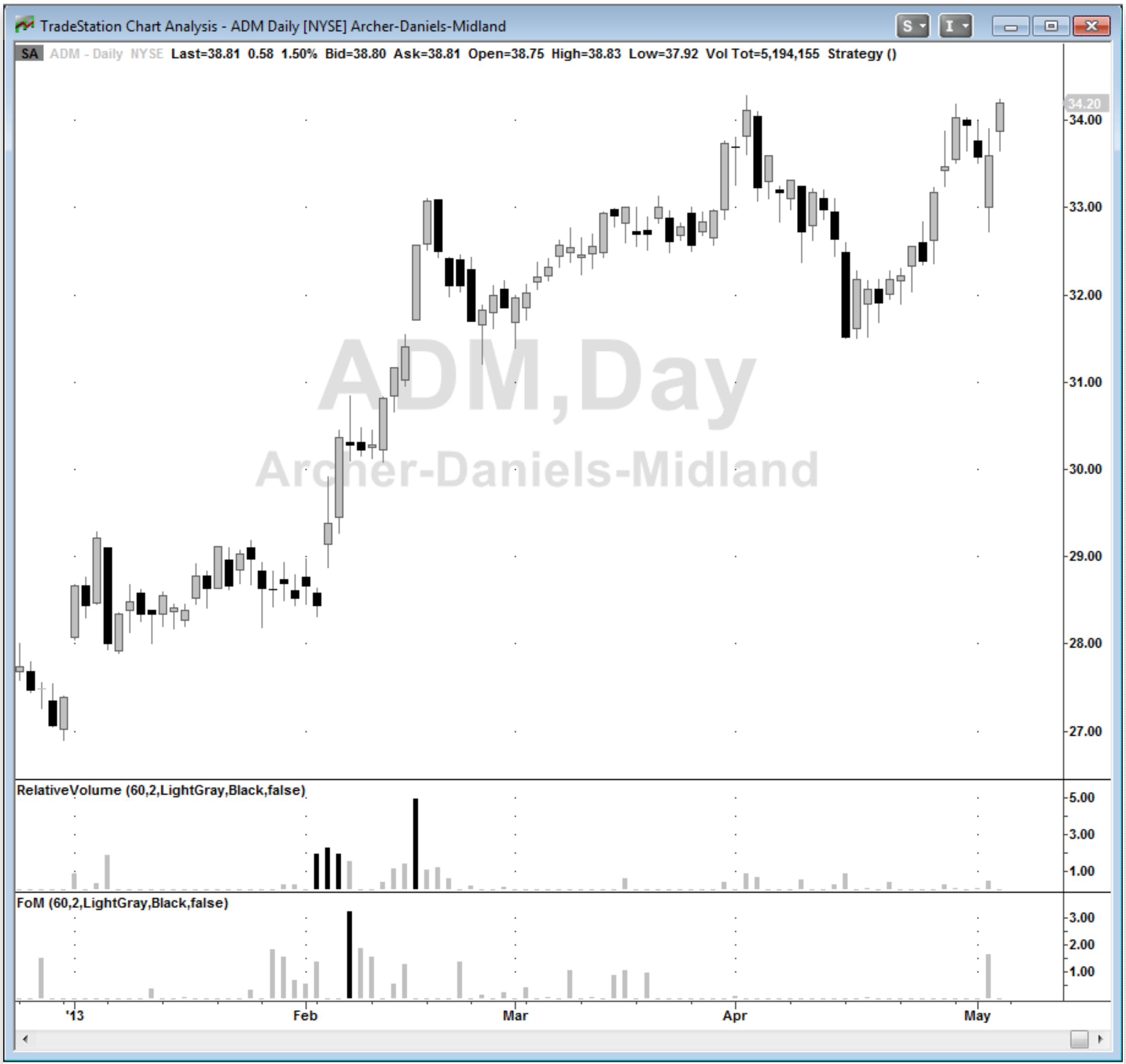
FoM (Indicator)

```
{Reference: Technical Analysis
of Stocks & Commodities, Apr 2014.
Article: Evidence-Based Support & Resistance
Indicator Name : FoM
}
inputs:
        int Period( 60 ),
        double NumStdDevs( 2 ),
        int FoMBelowSdevColor( LightGray ),
        int FoMAboveSdevColor( Black ),
        bool AllowNegativePlots( false ) ;
variables:
        double MyVolume( 0 ),
        double AvgVolume( 0 ),
        double StdDevVol( 0 ),
        double RelVolume( 0 ),
        double AMove( 0 ),
        double TheMin( 0 ),
        double TheMax( 0 ),
        double TheMove( 0 ),
        double TheMinVol( 0 ),
        double TheMaxVol( 0 ),
        double TheVol( 0 ),
        double VByM( 0 ),
        double AvF( 0 ),
        double SdF( 0 ),
        double TheFoM( 0 )
         ;

if BarType >= 2 and BarType < 5 then
        MyVolume = Volume
else
        MyVolume =      Ticks ;


AMove = AbsValue( ( Close - Close[1] ) / Close[1] ) ;
TheMin = Lowest( AMove, Period ) ;
TheMax = Highest( AMove, Period ) ;

if ( TheMax - TheMin ) > 0 then
        TheMove = 1 + ( ( AMove - TheMin )
        * ( 10 - 1 ) ) / ( TheMax - TheMin ) ;

AvgVolume = Average( MyVolume, Period ) ;
StdDevVol = StdDev( MyVolume, Period ) ;
RelVolume = iff( StdDevVol <> 0 ,
( MyVolume - AvgVolume ) / StdDevVol, 0 ) ;

TheMinVol = Lowest( RelVolume, Period ) ;
TheMaxVol = Highest( RelVolume, Period ) ;

if ( TheMaxVol - TheMinVol ) > 0 then
        TheVol = 1 + ( ( RelVolume - TheMinVol )
        * ( 10 - 1 ) ) / ( TheMaxVol - TheMinVol ) ;

VByM = iff( TheMove <> 0, TheVol / TheMove , 0 ) ;
AvF = Average( VByM, Period ) ;
Sdf = StdDev( VByM, Period ) ;
TheFoM = iff( SdF <> 0, ( VByM - AvF ) / SdF, 0 ) ;

if AllowNegativePlots = false then
        TheFoM = MaxList( TheFom, 0 ) ;

Plot1( TheFoM, "FoM" ) ;

if TheFoM < NumStdDevs then
        SetPlotColor( 1, FoMBelowSdevColor )
else
        SetPlotColor( 1, FoMAboveSdevColor ) ;
```

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-2014. The ELD filename is “_TASC_EVIDENCEBASED_SR.ELD.”

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

A sample chart is shown in Figure 1.



TradeStation Chart Analysis - ADM Daily [NYSE] Archer-Daniels-Midland

SA ADM - Daily NYSE Last=38.81 0.58 1.50% Bid=38.80 Ask=38.81 Open=38.75 High=38.83 Low=37.92 Vol Tot=5,194,155 Strategy ()

RelativeVolume (60,2,LightGray,Black,false)

FoM (60,2,LightGray,Black,false)

Created with TradeStation. ©TradeStation Technologies, Inc. All rights reserved.

FIGURE 1: TRADESTATION. Here is a daily chart of ADM with the RelativeVolume and FoM indicators inserted.

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

METASTOCK: APRIL 2014

Melvin Dickover’s article in this issue, “Evidence-Based Support & Resistance,” introduces two new volume-based indicators. The MetaStock formulas for the indicators are as follows:

```
Relative volume indicator
x:= Input("Standard Deviation Periods", 5, 120, 60);
y:= Input("Number of Deviations", 0, 4, 2);
av:= Mov(V, x, S);
sd:= Std(V, x);
relVol:= (V-av)/sd;

If(relVol>y, relvol, 0);
If(relvol<=y, If(relvol>0,relvol, 0),0);

 Freedom of Movement (FoM) indicator
x:= Input("Standard Deviation Periods", 5, 120, 60);
y:= Input("Number of Deviations", 0, 4, 2);
av:= Mov(V, x, S);
sd:= Std(V, x);
relVol:= (V-av)/sd;
aMove:= Abs((C-Ref(C,-1))/Ref(C,-1));
theMin:= LLV(aMove,x);
theMax:= HHV(aMove,x);
denom:= If(theMax-theMin = 0, -1, theMax-theMin);
theMove:= 1 +((aMove-theMin)*(10-1))/Abs(denom);
MinV:= LLV(relVol,x);
MaxV:= HHV(relVol,x);
denomv:= If(MaxV-MinV = 0, -1, MaxV-MinV);
theVol:= 1 + ((relVol-MinV)*(10-1))/Abs(denomV);
vByM:= theVol/theMove;
avF:= Mov(vByM, x, S);
sdF:= Std(vByM, x);
theFoM:= (vByM-avF)/sdF;

If(theFoM>y, theFoM, 0);
If(theFoM<=y, If(theFoM>0, theFoM, 0),0);
```

The last two lines of both formulas allow them to be plotted on a chart in two colors. After plotting the indicator, you will see two lines. One at a time, select a line. Then use the color and style toolbars in the bottom-left corner of your MetaStock screen to change the line style to *histogram*. You can also change the color and weight (thickness) of the line.

—William Golson
MetaStock Technical Support
www.metastock.com

BACK TO LIST

THINKORSWIM: APRIL 2014

In “Evidence-Based Support & Resistance,” author Melvin Dickover describes how to find support and resistance areas by using more than just price levels. Dickover provides evidence for using volume-based price levels for identifying these classic technical analysis areas.

Based on the article, we have created two new studies for thinkorswim users in our proprietary scripting language, thinkScript. To load them, simply click on each of these links: http://tos.mx/yzefQi and http://tos.mx/sL8TVa. You can adjust their parameters within the *edit studies* window to fine-tune your variables.

In Figure 2, you see a three-month daily chart of Archer-Daniels Midland (ADM). We have drawn support and resistance levels on the price chart by using the logic described in Dickover’s article. In order to draw all necessary lines, click on the links, save each in thinkorswim, and rename them. The first link, http://tos.mx/yzefQi, should be renamed FreedomOfMovment or FoM and the second, http://tos.mx/sL8TVa, should be renamed RelativeVolumeStDev.

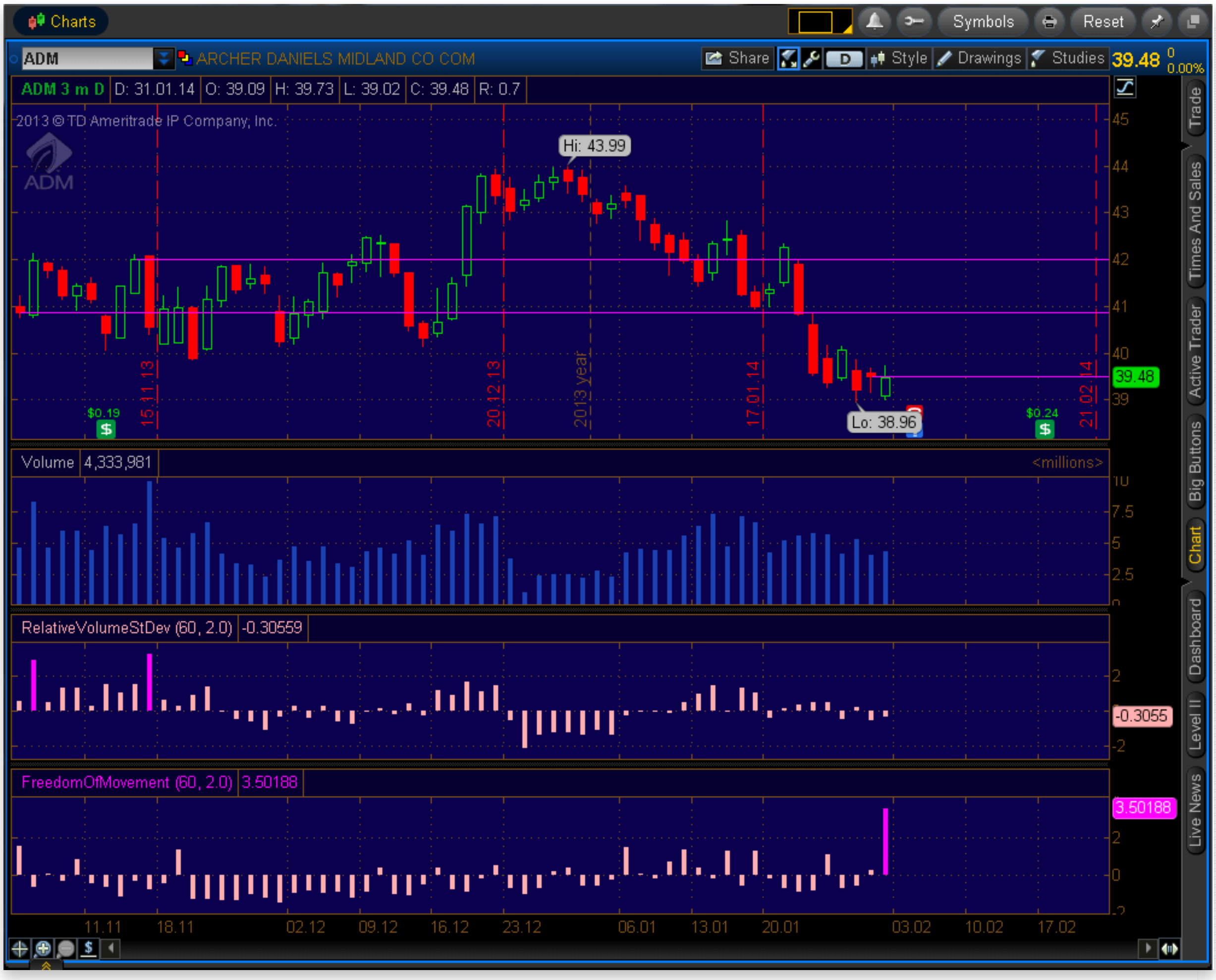


FIGURE 2: THINKORSWIM. Here is a sample three-month daily chart of Archer-Daniels Midland (ADM) with the relative volume and Freedom of Movement indicators in the lower panes.

Next, use the price level drawing tool, found in thinkorswim charts under the *drawings* menu at the top right, to draw your price levels based on the spikes on your studies. You can expand the chart area to show time to the right by clicking on *style* then *settings* and the *time axis* tab. Now your drawings are looking forward and displaying future support and resistance lines.

If you have questions or comments, please email us at support@thinkorswim.com and mention *Technical Analysis of* STOCKS & COMMODITIES magazine.

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

BACK TO LIST

WEALTH-LAB: APRIL 2014

In “Evidence-Based Support & Resistance” in this issue, author Melvin Dickover presents a new framework for discovering supply and demand pools and creating dependable support and resistance lines that claim to identify the prices where those pools are located on the chart. The underlying method is quite simple to grasp.

Because drawing the Defended Price Lines (DPLs) is to some extent a heuristic process, the example strategy plots the DPLs using a rudimentary algorithm: a line is drawn when either the relative volume or Freedom of Movement jumps two or more standard deviations away from the mean. We recommend using that only as a starting point, though. To avoid getting distracted by too many DPLs, pinpoint the most influential ones by considering such criteria as clustering behavior, strong momentum, and high volume.

To hide the complexity and make the process easier, the two new indicators (relative volume and Freedom of Movement) have been added to the Wealth-Lab TASCIndicators library. To execute the included sample code, Wealth-Lab users need to install (or update) 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.

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 MyStrategy : WealthScript
        {
                protected override void Execute()
                {
                        int period = 60;
                        double d = 2;
                        DataSeries relVol = Close*0;    relVol.Description = 
"Relative Volume";
                        DataSeries theFoM = Volume*0;   theFoM.Description = "Freedom 
of Movement";

                        for(int bar = 0; bar < Bars.Count; bar++)
                        {
                                relVol[bar] = RelVol.Series(Bars,period)[bar] > 0 ? 
RelVol.Series(Bars,period)[bar] : 0;
                                theFoM[bar] = FOM.Series(Bars,period)[bar] > 0 ? 
FOM.Series(Bars,period)[bar] : 0;
                        }

                        HideVolume();
                        ChartPane r = CreatePane( 30,false,true );
                        PlotSeries( r, relVol, Color.Black, LineStyle.Histogram, 3 );
                        ChartPane f = CreatePane( 30,false,true );
                        PlotSeries( f, theFoM, Color.Black, LineStyle.Histogram, 3 );

                        for(int bar = 1; bar < Bars.Count; bar++)
                        {
                                if (relVol[bar] <= d)
                                        SetSeriesBarColor( bar, relVol, 
Color.DarkGray);
                                if (theFoM[bar] < d)
                                        SetSeriesBarColor( bar, theFoM, 
Color.DarkGray);

                                if (theFoM[bar] > d || relVol[bar] > d) // Draw DPLs
                                {
                                        DrawLine(PricePane,bar,Close[bar],Bars.Count-
1,Close[bar],Color.Blue,LineStyle.Solid,1);
                                }
                        }
                }
        }
}
```

A sample chart is shown in Figure 3.

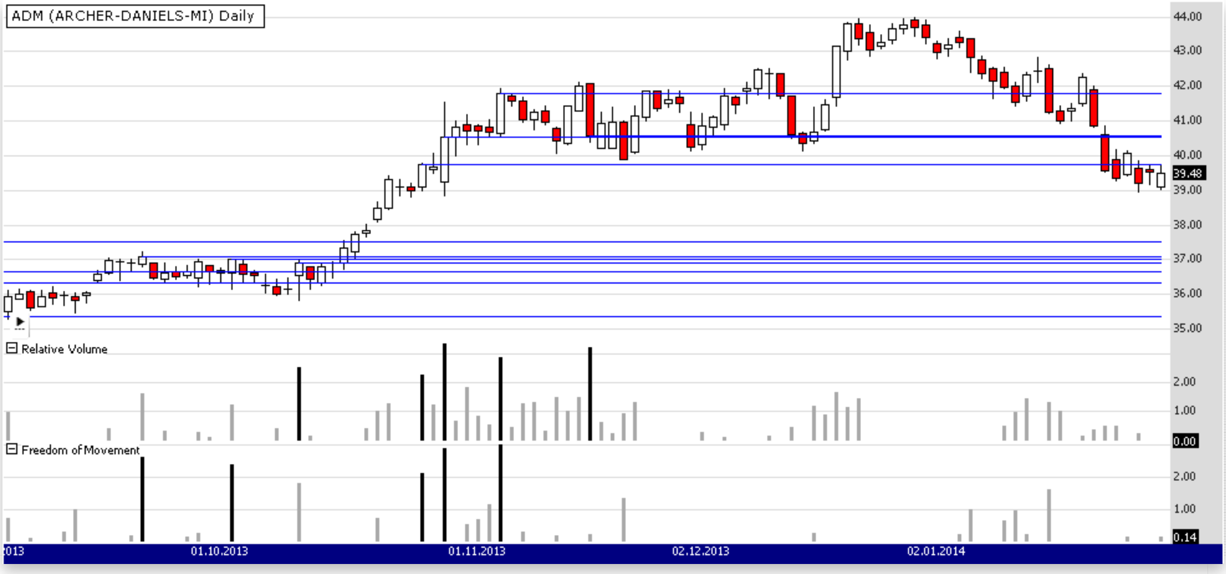


FIGURE 3: WEALTH-LAB. This sample Wealth-Lab 6 chart illustrates how the DPLs can be plotted following some of the rules from Melvin Dickover's article in this issue. The DPLs are the blue lines on a daily chart of ADM. The relative volume and Freedom of Movement (FoM) indicators are plotted on the bottom pane for reference purposes.

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

BACK TO LIST

NEUROSHELL TRADER: APRIL 2014

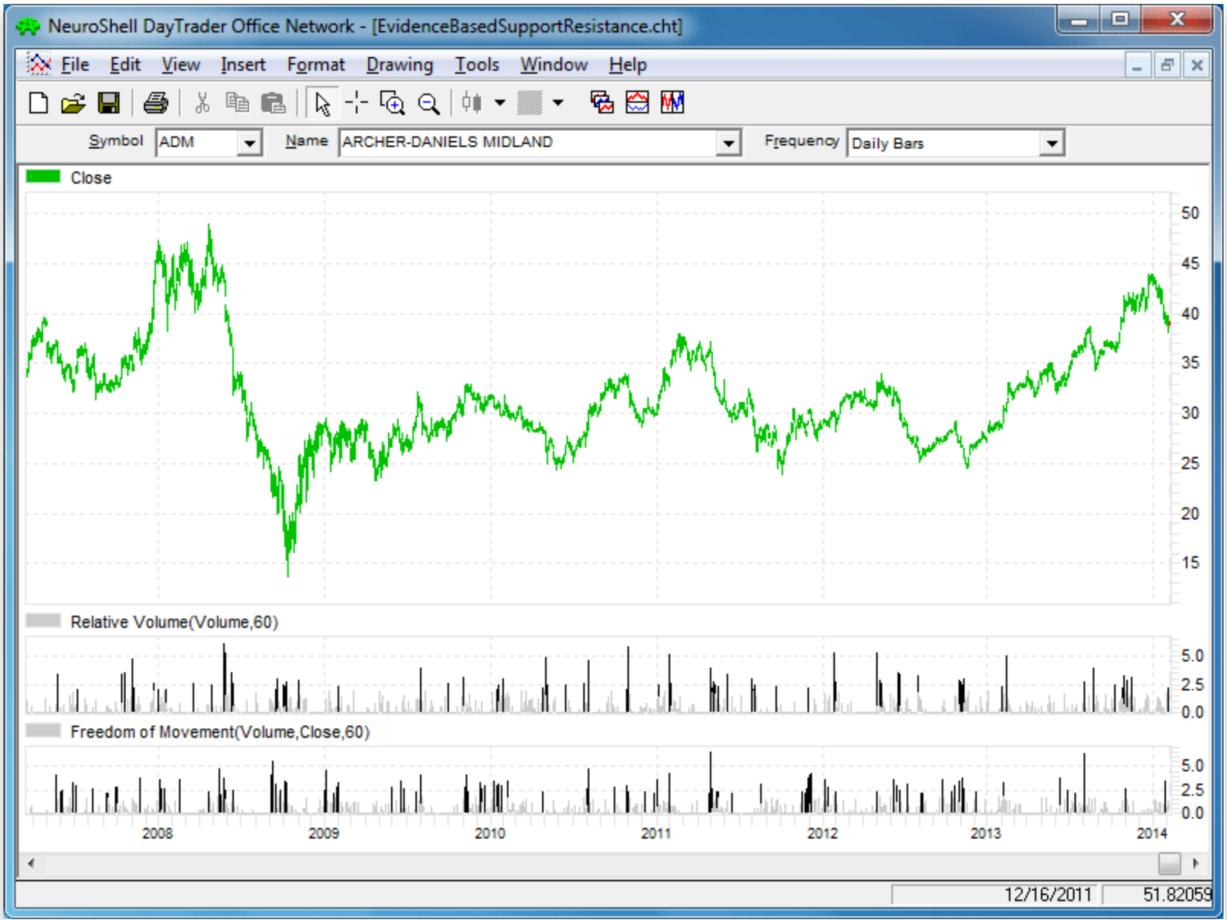
The relative volume and Freedom of Movement indicators described by Melvin Dickover in his article in this issue, "Evidence-Based Support & Resistance." can be easily implemented with a few of NeuroShell Trader's 800+ indicators. Simply select "New indicator" from the Insert menu and use the indicator wizard to recreate the following indicators:

```
Relative Volume:        StndNormZScore( Volume,  PERIOD )

RelVol Significance:    A>B( Relative Volume, STDEVS )

FoM:    StndNormZScore( Divide(Add2( 1, Mult2( SimpleStoch%K(
StndNormZScore(Volume,PERIOD), PERIOD), 0.09 ), Add2( 1, Mult2( SimpleStoch%K( abs(
PercentChange(Close,1)), PERIOD), 0.09)), PERIOD)
```

```
FoM Significance:       A>=B( FoM, STDEVS )
```

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.


NeuroShell DayTrader Office Network - [EvidenceBasedSupportResistance.cht]
File
Edit
View
Insert
Format
Drawing
Tools
Window
Help
Close
50
45
40
35
30
25
20
15
Relative Volume(Volume,60)
5.0
2.5
0.0
Freedom of Movement(Volume,Close,60)
5.0
2.5
0.0
2008
2009
2010
2011
2012
2013
2014

Symbol ADM
Name ARCHER-DANIELS MIDLAND
Frequency Daily Bars

12/16/2011 51.82059

FIGURE 4: NEUROSHELL TRADER. This NeuroShell Trader chart displays the relative volume and Freedom of Movement indicators based on Melvin Dickover's article in this issue.

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

BACK TO LIST

AIQ: APRIL 2014

The AIQ code based on Melvin Dickover's article in this issue, "Evidence-Based Support & Resistance," can be found at http://www.TradersEdgeSystems.com/traderstips.htm.

The code provided there and also shown here will allow AIQ users to plot the two indicators presented by Dickover, the relative volume and Freedom of Movement (FoM) indicators, which are shown in Figure 5.

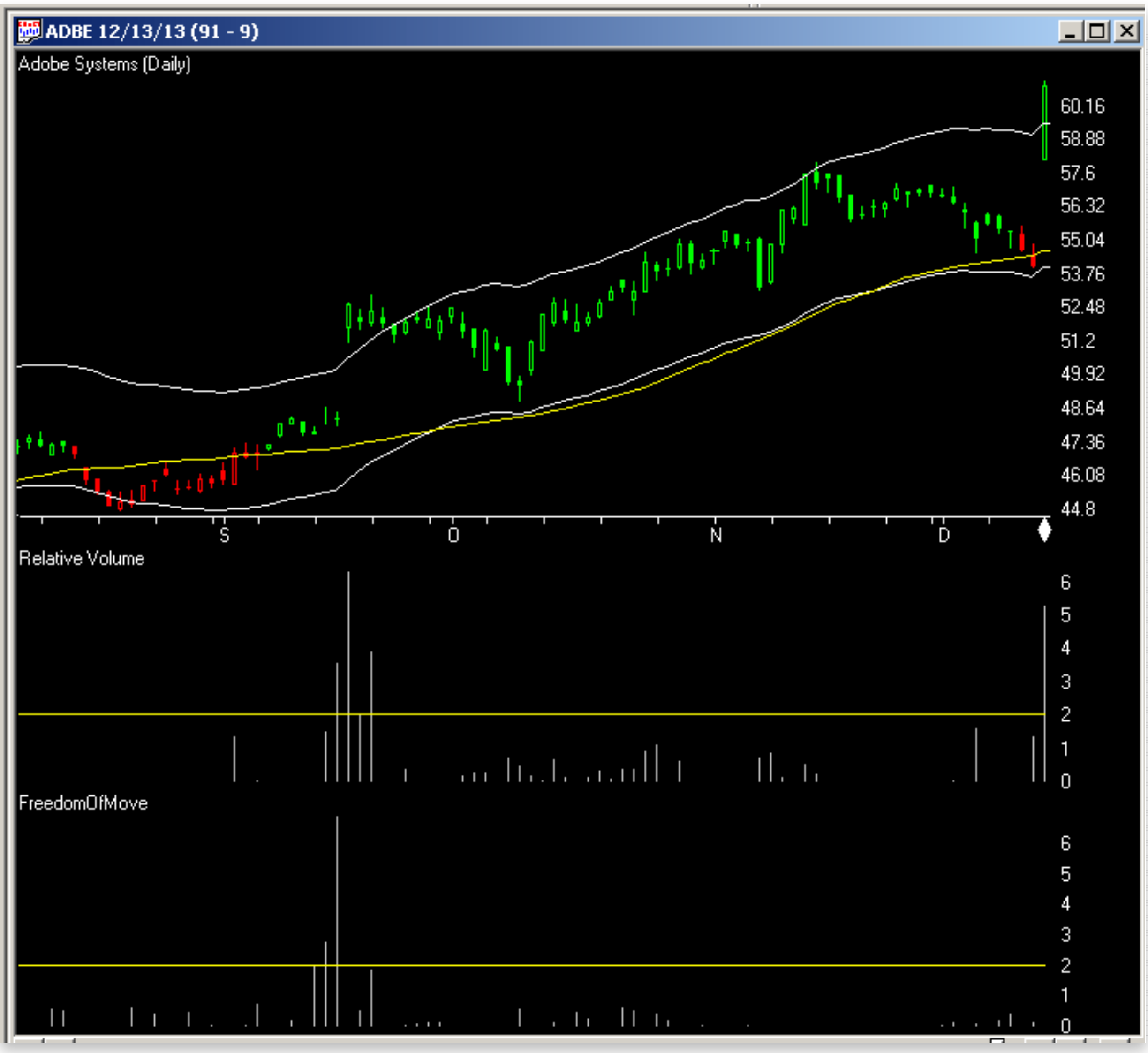


FIGURE 5: AIQ. Here is a sample plot of the relative volume and Freedom of Movement indicators on a chart of Adobe Systems (ADBE). The horizontal lines

are set at two standard deviations.

```
!EVIDENCE-BASED SUPPORT & RESISTANCE
!Author: Melvin E. Dickover, TASC April 2014
!Coded by: Richard Denning, 2/5/2014
!www.TradersEdgeSystems.com

!INPUTS:
vLen is 60.
sdSigLvl is 2.

V is [volume].
C is [close].
C1 is valresult(C,1).

avgV is simpleavg(V,vLen).
sdV is sqrt(variance(V,vLen)).
rv is (V-avgV)/sdV.
rvi is iff(rv>0,rv,0).    !PLOT

aMove is abs((C-C1)/C1).
minM is lowresult(aMove,vLen).
maxM is highresult(aMove,vLen).
normM is 1+((aMove-minM)*(10-1))/(maxM-minM).

minV is lowresult(rv,vLen).
maxV is highresult(rv,vLen).
normV is 1+((rv-minV)*(10-1))/(maxV-minV).

vByM is normV/normM.
avgF is simpleavg(vByM,vLen).
sdF is sqrt(variance(vByM,vLen)).
FofM is (vByM-avgF)/sdF.
FofMI is iff(FofM>0,FofM,0).!PLOT
```

—Richard Denning
info@TradersEdgeSystems.com
for AIQ Systems

BACK TO LIST

TRADERSSTUDIO: APRIL 2014

The TradersStudio code based on Melvin Dickover’s article in this issue, “Evidence-Based Support & Resistance,” is provided at both of 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 MED_NORMALIZE: Returns a normalized value that can be used to make comparisons
- Function MED_RELATIVE_VOLUME: Returns the values for the relative volume indicator
- Function MED_FREEEDOM_MOVE: Returns the values from the Freedom of Movement (FoM) indicator
- Indicator MED_RELATIVE_VOLUME_IND: Indicator plot for the relative volume indicator
- Indicator MED_FREEDOM_MOVE_IND: Indicator plot for the Freedom of Movement (FoM) indicator

```
'EVIDENCE-BASED SUPPORT & RESISTANCE
'Author: Melvin E. Dickover, TASC April 2014
'Coded by: Richard Denning, 2/7/2014
'www.TradersEdgeSystems.com

'Normalizes input variable so that comparison can be made
Function MED_NORMALIZE(notN as bararray,lenN)
Dim norm As BarArray
If (Highest(notN,lenN)-Lowest(notN,lenN)) <> 0 Then
    norm = 1+((notN-Lowest(notN,lenN))*(10-1))/(Highest(notN,lenN)-Lowest(notN,lenN))
End If
MED_NORMALIZE = norm
End Function
'------------------------------------------------
Function MED_RELATIVE_VOLUME(lenV)
Dim sdV As BarArray
Dim RV As BarArray
sdV = StdDev(V,lenV,0)
If sdV<>0 Then RV = (V-Average(V,lenV))/sdV
MED_RELATIVE_VOLUME = RV
End Function
'------------------------------------------------
'Returns the Fredom Of Movement values
Function MED_FREEDOM_MOVE(vLen)
Dim normM As BarArray
Dim normV As BarArray
Dim vByM As BarArray
Dim avgF As BarArray
normM = MED_NORMALIZE(Abs(C/C[1]-1),vLen)
normV = MED_NORMALIZE(MED_RELATIVE_VOLUME(vLen),vLen)
If normM<>0 Then vByM = normV / normM
avgF = Average(vByM,vLen)
MED_FREEDOM_MOVE = (vByM-avgF)/StdDev(vByM,vLen)
End Function
'------------------------------------------------
'Indicator plot for Relative Volume indicator
Sub MED_RELATIVE_VOLUME_IND(vLen)
Dim RV As BarArray
Dim RVI As BarArray
RV = MED_RELATIVE_VOLUME(vLen)
RVI = Max(RV,0)
plot1(RVI)
plot2(2)
'Print FormatDateTime(Date)," ",RVI
End Sub
'------------------------------------------------
```

```
'Indicator plot for Freedom Of Movement indicator
Sub MED_FREEDOM_MOVE_IND(vLen)
Dim FoM As BarArray
Dim FoMI As BarArray
FoM = MED_FREEDOM_MOVE(vLen)
FoMI = max(FoM,0)
plot1(FoMI)
plot2(2)
'Print FormatDateTime(Date)," ",FoMI
End Sub
'------------------------------------------------
```

In Figure 6, I show Dickover's two indicators, Freedom of Movement and relative volume, on a chart of Alaska Air (ALK). The horizontal lines are set at two standard deviations.

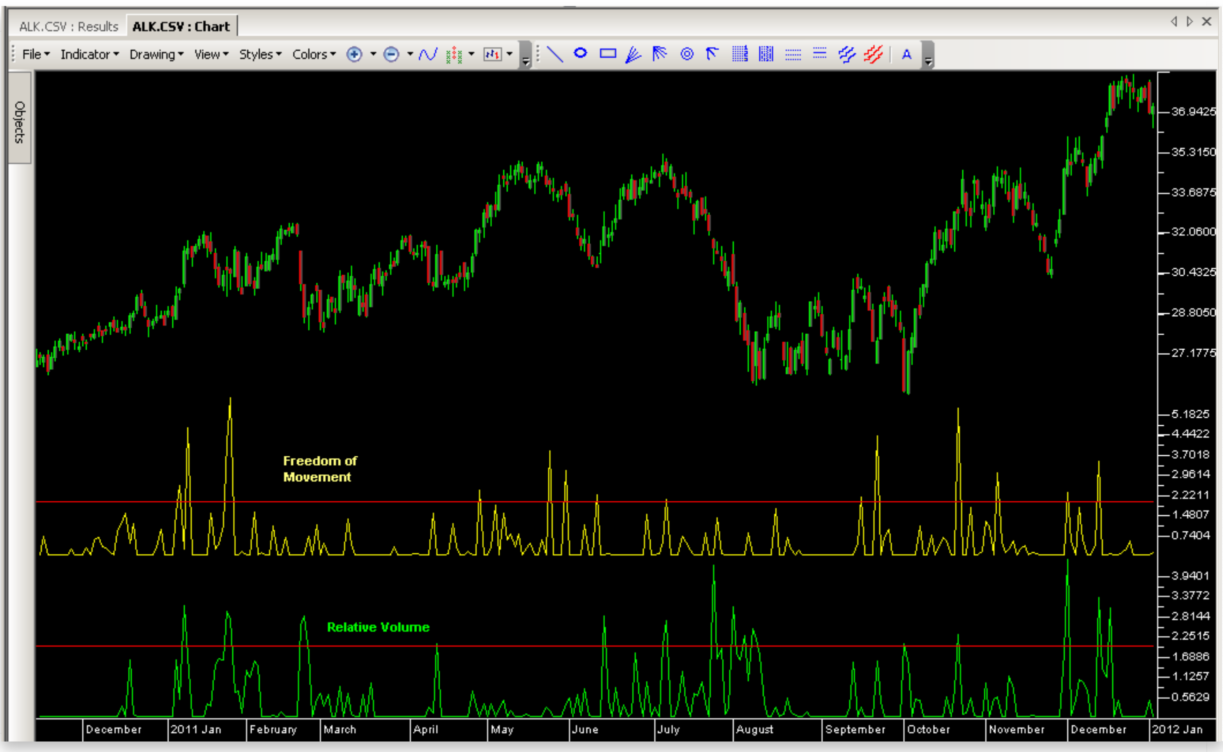


FIGURE 6: TRADERSSTUDIO. Here, the Freedom of Movement (yellow) and relative volume (green) indicators are shown on a chart of Alaska Air (ALK).

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

NINJATRADER: APRIL 2014

We have implemented the RelativeVolume and Freedom of Movement (FoM) indicators that are presented in Melvin Dickover's article in this issue, "Evidence-Based Support & Resistance." These indicators are available for download at www.ninjatrader.com/SC/April2014SC.zip.

Once they have 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 both indicators' source code by selecting the menu Tools → Edit NinjaScript → Strategy from within the NinjaTrader Control Center window and selecting the RelativeVolume and FoM files.

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

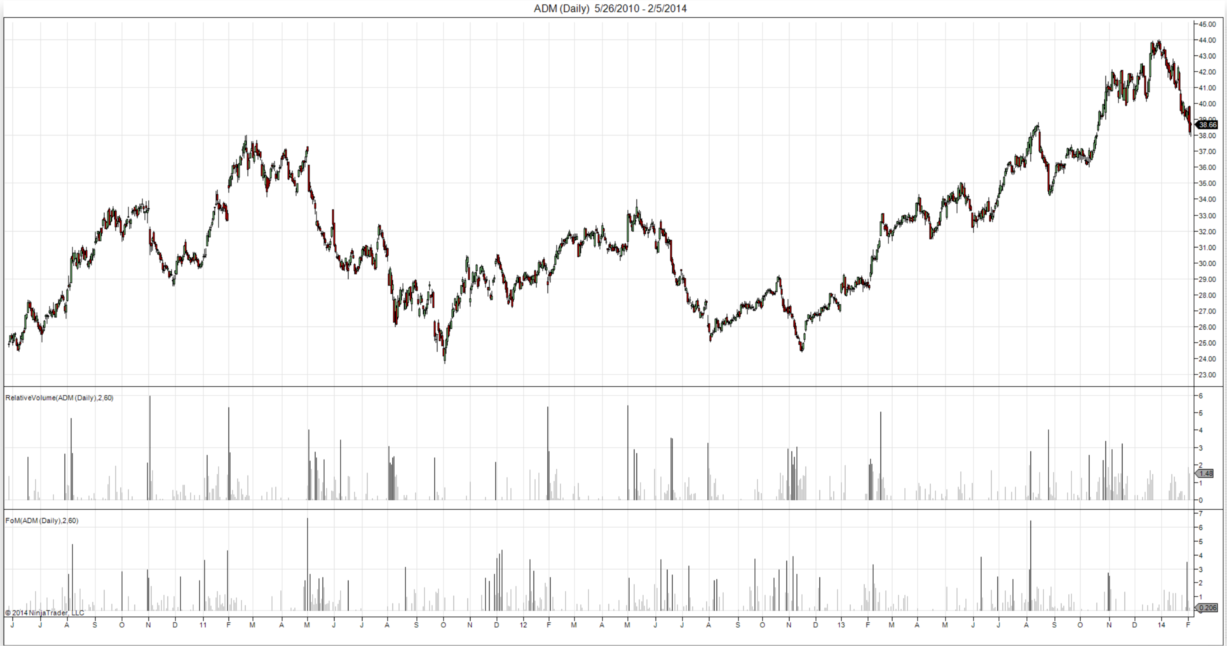


FIGURE 7: NINJATRADER. This screenshot shows the RelativeVolume and FoM indicators applied to a daily chart of Archer-Daniels Midland (ADM) stock.

—Raymond Deux & Brandon Sutrina
NinjaTrader, LLC
www.ninjatrader.com

BACK TO LIST

UPDATA: APRIL 2014

This Traders' Tip is based on Melvin Dickover's article in this issue, "Evidence-Based Support & Resistance."

The author develops two indicators based on price and volume behavior to better determine support and resistance lines. The first is a relative volume indicator, which identifies large-volume days, illustrated with a black bar in an otherwise gray histogram. The second is the Defended Price Line (DPL).Updata has slightly extended the article by adding rules for the additional DPL, plotted at the close price after a cluster of large-volume black bars.

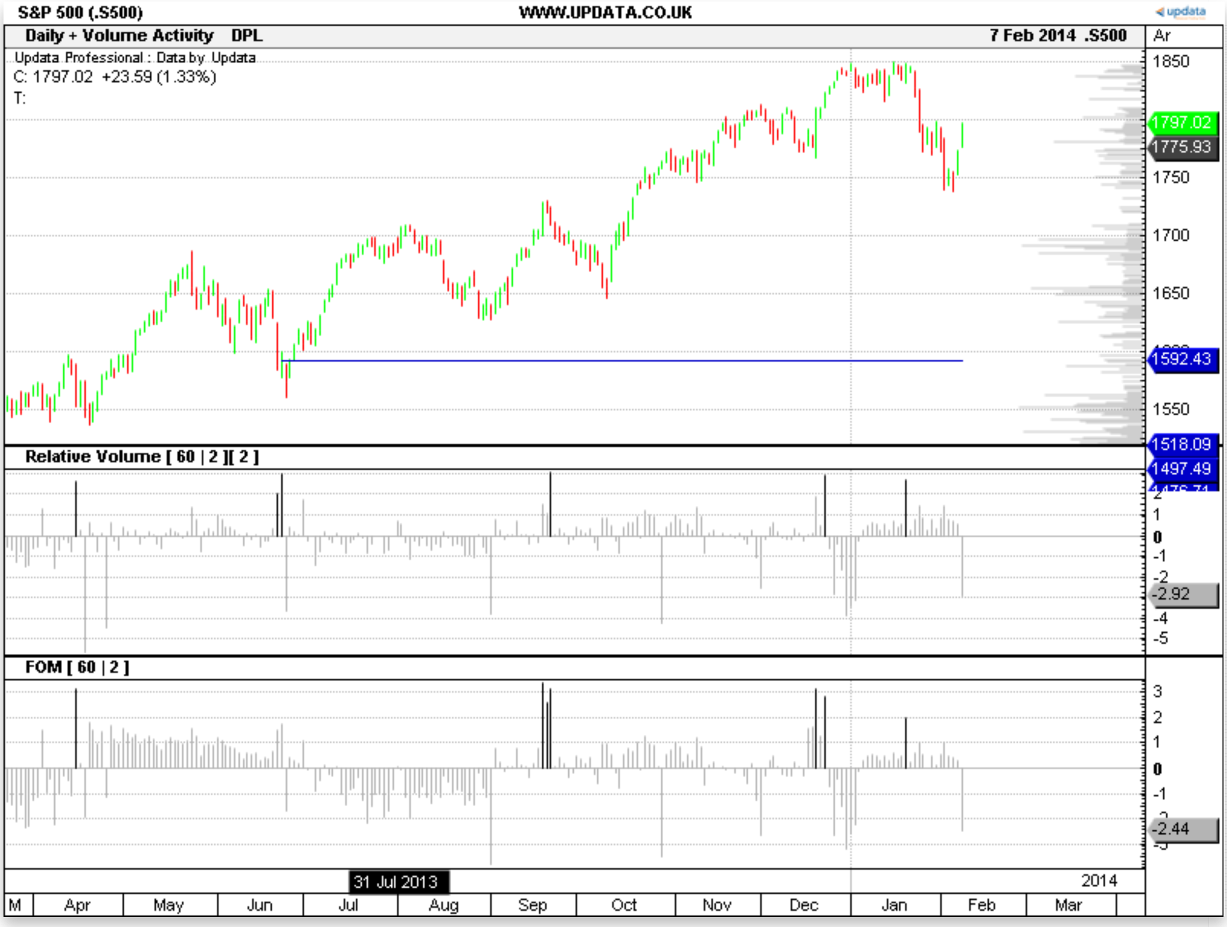


FIGURE 8: UPDATA. Here, the relative volume [60|2] and Freedom of Movement [60|2] indicators are applied to daily S&P 500 index data. The DPLs are drawn with relative volume clusters of size 2.

The second colored histogram (Figure 8) is Dickover's Freedom of Movement (FoM) indicator, which seeks to identify when price movement has suddenly become restricted by comparing a price move to volume change over a period.

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.

```
'Relative Volume (Cluster)
PARAMETER "Period" #PERIOD=60
PARAMETER "Std. Dev." @STDDEV=2
PARAMETER "Cluster Size" #CLUSTER=3
NAME "DPL" ""
NAME2 "Relative Volume [" #PERIOD "|" @STDDEV "][" #CLUSTER "]" ""
DISPLAYSTYLE 2LINES
INDICATORTYPE TOOL
INDICATORTYPE2 CHART
PLOTSTYLE2 HISTOGRAM
@AVGVOL=0
@STDDEVVOL=0
@RELATIVEVOL=0
#DISTANCE=0
@VOLTHRESH=0
@VOLTHRESHSUM=0
@CLUSTERFOUND=0
FOR #CURDATE=#PERIOD TO #LASTDATE
   @AVGVOL=SGNL(VOL,#PERIOD,M)
   @STDDEVVOL=STDDEV(VOL,#PERIOD)
   @RELATIVEVOL=(VOL-@AVGVOL)/@STDDEVVOL
   #DISTANCE=#LASTDATE-#CURDATE
   '=1 If Relative Vo greater then Std. Dev. Parameter,=0 otherwise
   @VOLTHRESH=@RELATIVEVOL>@STDDEV
   @VOLTHRESHSUM=SGNL(@VOLTHRESH,#CLUSTER,M)*#CLUSTER
   'Tests if condition met for cluster definition by summing variable
   @CLUSTERFOUND=0
   IF @VOLTHRESHSUM>=#CLUSTER
      @CLUSTERFOUND=1
   ENDIF
   'Plot Relative Vol Histogram
   IF @RELATIVEVOL>@STDDEV
      @PLOT2=@RELATIVEVOL
      COLOUR2 RGB(0,0,0)
   ELSE
      @PLOT2=@RELATIVEVOL
      COLOUR2 RGB(180,180,180)
   ENDIF
   'Plot DPL if more at least '3' black bars clustered consecutively
   IF @CLUSTERFOUND=1
      COLOUR RGB(0,0,200)
      DRAWLINE 0,CLOSE,-#DISTANCE,CLOSE
   ENDIF
NEXT
```

```
'Freedom Of Movement
PARAMETER "Period" #PERIOD=60
PARAMETER "Std. Dev." @STDDEV=2
DISPLAYSTYLE LINE
INDICATORTYPE CHART
PLOTSTYLE HISTOGRAM
NAME "FOM [" #PERIOD "|" @STDDEV "]" ""
@MOVE=0
@MINMOVE=0
@MAXMOVE=0
@AVERAGEMOVE=0
@STDDEVVOL=0
@AVERAGEVOL=0
@RELATIVEVOLUME=0
@MAXVOLUME=0
```

```
@MINVOLUME=0
@VOL=0
@VOLUMEBYMOVE=0
@AVGVOLUMEBYMOVE=0
@AVERAGEFREEDOM=0
@STDDEVFREEDOM=0
@FOM=0
FOR #CURDATE=#PERIOD TO #LASTDATE
    'Movement
    @MOVE=ABS((CLOSE-CLOSE(1))/CLOSE)
    @MAXMOVE=PHIGH(@MOVE,#PERIOD)
    @MINMOVE=PLOW(@MOVE,#PERIOD)
    IF (@MAXMOVE-@MINMOVE)>0
       @MOVE=1+((@MOVE-@MINMOVE)*(10-1))/(@MAXMOVE-@MINMOVE)
    ENDIF
    'Compute Relative Volume
    @AVERAGEVOL=SGNL(VOL,#PERIOD,M)
    @STDDEVVOL=STDDEV(VOL,#PERIOD)
    @RELATIVEVOLUME=(VOL-@AVERAGEVOL)/@STDDEVVOL
    @MAXVOLUME=PHIGH(@RELATIVEVOLUME,#PERIOD)
    @MINVOLUME=PLOW(@RELATIVEVOLUME,#PERIOD)
    IF (@MAXVOLUME-@MINVOLUME)>0
       @VOL=1+((@RELATIVEVOLUME-@MINVOLUME)*(10-1))/(@MAXVOLUME-@MINVOLUME)
    ENDIF
    'Compute Freedom of Movement in StdDevs
    @VOLUMEBYMOVE=@VOL/@MOVE
    @AVERAGEFREEDOM=SGNL(@VOLUMEBYMOVE,#PERIOD,M)
    @STDDEVFREEDOM=STDDEV(@VOLUMEBYMOVE,#PERIOD)
    @FOM=(@VOLUMEBYMOVE-@AVERAGEFREEDOM)/@STDDEVFREEDOM
    IF @FOM<@STDDEV
       @PLOT=@FOM
       COLOUR RGB(180,180,180)
    ELSE
       @PLOT=@FOM
       COLOUR RGB(0,0,0)
    ENDIF
NEXT
```

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

BACK TO LIST

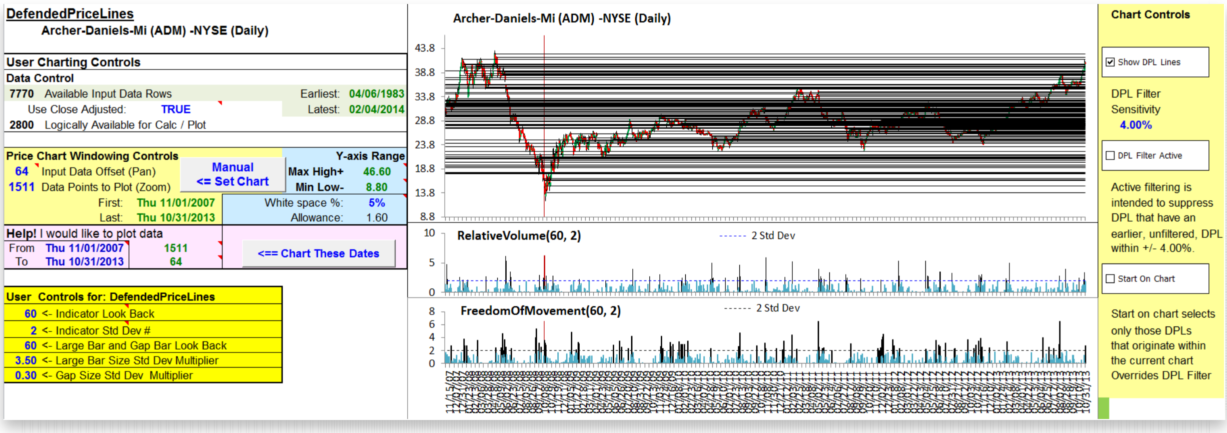
MICROSOFT EXCEL: APRIL 2014

In his article in this issue, “Evidence-Based Support & Resistance,” author Melvin Dickover provides two new indicators that point out locations of possible support and resistance. With these indicators and a simple set of rules, Dickover determines initial support or resistance levels, which he manually plots as Defended Price Lines (DPLs).

Since the indicators show repeated support and resistance levels that tend to cluster near the same levels, Dickover adjusts his DPL up or down to reflect this new confirming information with a single best-fit for the apparent localized support or resistance level.

What you see in my Figure 9 is the same data that's shown in Figure 1 of Dickover's article. I have used the heuristics described in his article to have Excel set the DPLs.

As you can see, there are lots of them. Many originate off the chart to the left. The darker areas show critical price point confirmation by way of the closely located DPLs.



DefendedPriceLines
Archer-Daniels-Mi (ADM) -NYSE (Daily)

User Charting Controls
Data Control
7770 Available Input Data Rows Earliest: 04/06/1983
Use Close Adjusted: TRUE Latest: 02/04/2014
2800 Logically Available for Calc / Plot

Price Chart Windowing Controls
64 Input Data Offset (Pan)
1511 Data Points to Plot (Zoom)
First: Thu 11/01/2007
Last: Thu 10/31/2013
Y-axis Range
Max High+ 46.60
Min Low- 8.80
White space %: 5%
Allowance: 1.60

Help! I would like to plot data
From Thu 11/01/2007 1511
To Thu 10/31/2013 64

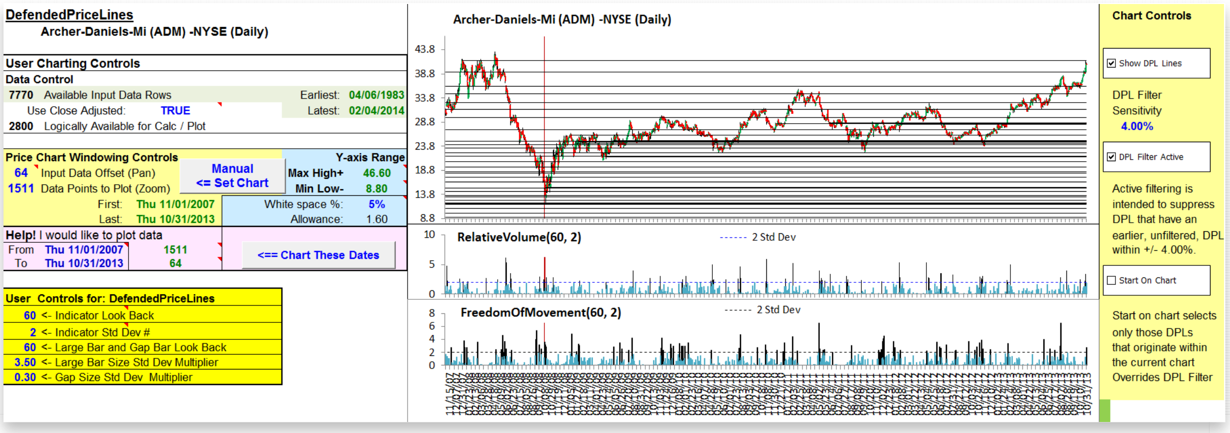
User Controls for: DefendedPriceLines
60 <- Indicator Look Back
2 <- Indicator Std Dev #
60 <- Large Bar and Gap Bar Look Back
3.50 <- Large Bar Size Std Dev Multiplier
0.30 <- Gap Size Std Dev Multiplier

Chart Controls
☑ Show DPL Lines
DPL Filter Sensitivity 4.00%
☐ DPL Filter Active
Active filtering is intended to suppress DPL that have an earlier, unfiltered, DPL within +/- 4.00%.
☐ Start On Chart
Start on chart selects only those DPLs that originate within the current chart Overrides DPL Filter

FIGURE 9: EXCEL, ACTIVE FILTER. Use the active filter to remove some of the cluster clutter.

Figure 10 shows the results of one filtering method that can "thin the herd" a bit by suppressing the plot of DPLs whose levels are within a user-specified tolerance proximity of older, already displayed DPLs. The method I used is not as precise as I would like, so some clutter remains.

An Excel limitation on the maximum number of series per chart (256) causes some interesting things to happen on the chart. In Figure 10, the active filter has eliminated some of the clutter shown in the upper-right part of Figure 9. This made room within the series limitation for some older-origination DPLs to appear in Figure 10. Compare the DPLs that run through the sharp valley formation near the vertical cursor.



DefendedPriceLines
Archer-Daniels-Mi (ADM) -NYSE (Daily)

User Charting Controls
Data Control
7770 Available Input Data Rows Earliest: 04/06/1983
Use Close Adjusted: TRUE Latest: 02/04/2014
2800 Logically Available for Calc / Plot

Price Chart Windowing Controls
64 Input Data Offset (Pan)
1511 Data Points to Plot (Zoom)
First: Thu 11/01/2007
Last: Thu 10/31/2013
Y-axis Range
Max High+ 46.60
Min Low- 8.80
White space %: 5%
Allowance: 1.60

Help! I would like to plot data
From Thu 11/01/2007 1511
To Thu 10/31/2013 64

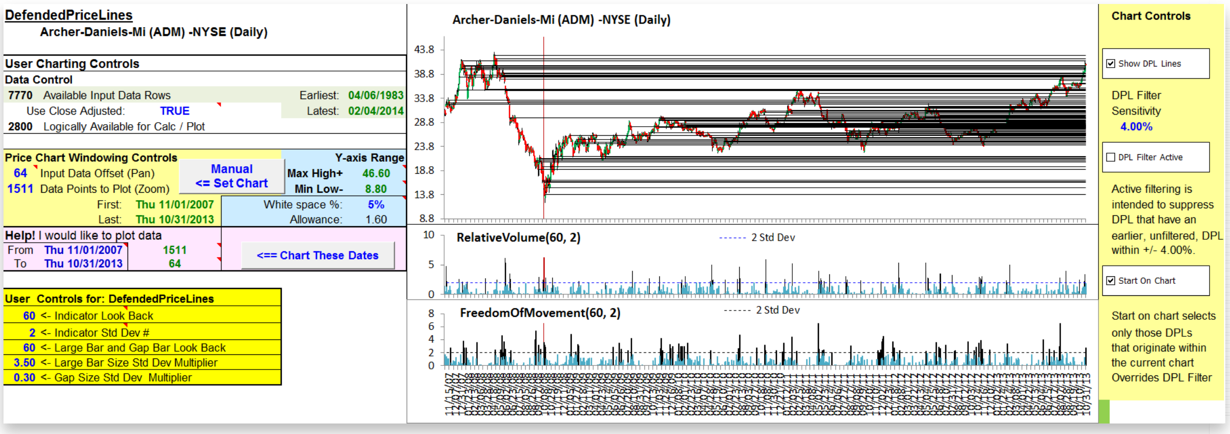
User Controls for: DefendedPriceLines
60 <- Indicator Look Back
2 <- Indicator Std Dev #
60 <- Large Bar and Gap Bar Look Back
3.50 <- Large Bar Size Std Dev Multiplier
0.30 <- Gap Size Std Dev Multiplier

Chart Controls
☑ Show DPL Lines
DPL Filter Sensitivity 4.00%
☑ DPL Filter Active
Active filtering is intended to suppress DPL that have an earlier, unfiltered, DPL within +/- 4.00%.
☐ Start On Chart
Start on chart selects only those DPLs that originate within the current chart Overrides DPL Filter

FIGURE 10: EXCEL, Defended Price Lines (DPLs). Here, we show only those DPLs that originate within

the current chart specifications.

Figure 11 does a better job of approximating Figure 1 from Dickover's article by restricting the display to DPLs that originate on the visible area of the chart. This version still gets congested as we move to the right. But it also makes clear the sort of manual DPL-level selections and adjustments that Dickover had to perform to arrive at his much-less cluttered Figure 1.



DefendedPriceLines
Archer-Daniels-Mi (ADM) -NYSE (Daily)

User Charting Controls
Data Control
7770 Available Input Data Rows — Earliest: 04/06/1983
Use Close Adjusted: TRUE — Latest: 02/04/2014
2800 Logically Available for Calc / Plot

Price Chart Windowing Controls
64 Input Data Offset (Pan)
1511 Data Points to Plot (Zoom)
First: Thu 11/01/2007
Last: Thu 10/31/2013
Y-axis Range
Max High+ 46.60
Min Low- 8.80
White space %: 5%
Allowance: 1.60

Help! I would like to plot data
From Thu 11/01/2007 1511
To Thu 10/31/2013 64

User Controls for: DefendedPriceLines
60 <- Indicator Look Back
2 <- Indicator Std Dev #
60 <- Large Bar and Gap Bar Look Back
3.50 <- Large Bar Size Std Dev Multiplier
0.30 <- Gap Size Std Dev Multiplier

Chart Controls
☑ Show DPL Lines
DPL Filter Sensitivity 4.00%
☐ DPL Filter Active
Active filtering is intended to suppress DPL that have an earlier, unfiltered, DPL within +/- 4.00%.
☑ Start On Chart
Start on chart selects only those DPLs that originate within the current chart Overrides DPL Filter

FIGURE 11: EXCEL, FREEDOM OF MOVEMENT AND DPLs. Volume and the FoM indicators are displayed with automatically generated DPLs.

Under the heading of user friendly, I have added a *follow* tab to my Traders' Tips Excel template.

Simply enter a list of Yahoo! Finance–acceptable symbols in column A of the *follow* tab. Then you can double-click on one of these symbols to automatically initiate data retrieval by way of the *InputPriceData* tab. Any time you retrieve data for a symbol listed on the *follow* tab, that row of the *follow* tab will be updated to record a *last trade* entry with time stamp. (See the notes on the *follow* tab.)

The spreadsheet file for this Traders' Tip can be downloaded here (NOTE: corrected version, posted 04-21-14). To successfully download it, follow these steps:

- Right-click on the Excel file link, then
- Select "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 April 2014 issue of
Technical Analysis of STOCKS & COMMODITIES magazine.

TRADERS' TIPS

May 2014

For this month's Traders' Tips, the focus is Donald W. Pendergast Jr.'s article in the 2014 Bonus Issue of Stocks & Commodities, "A Trading Method For The Long Haul." Here we present the May 2014 Traders' Tips code with possible implementations in various software.

The 2014 Bonus Issue of Stocks & Commodities magazine was mailed to paid subscribers in late February 2014. To get a copy of the Bonus Issue, you can become a subscriber to Stocks & Commodities magazine by visiting Traders.com or calling 800-TECHNICAL (800 832-4642).

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

TRADESTATION: MAY 2014

In "A Trading Method For The Long Haul" which appeared in the 2014 Bonus Issue of Stocks & Commodities, author Donald W. Pendergast Jr. discusses a method for selecting equities for possible trades. He then provides a set of rules for entering and exiting trades.

The author demonstrates a screening method using TradeStation's RadarScreen, which allows you to monitor up to 2,000 symbols in real time. The indicators he shows are all built in analysis techniques that are included with the TradeStation platform. As an alternative, the trader could use the TradeStation Scanner to search an even broader range of symbols.

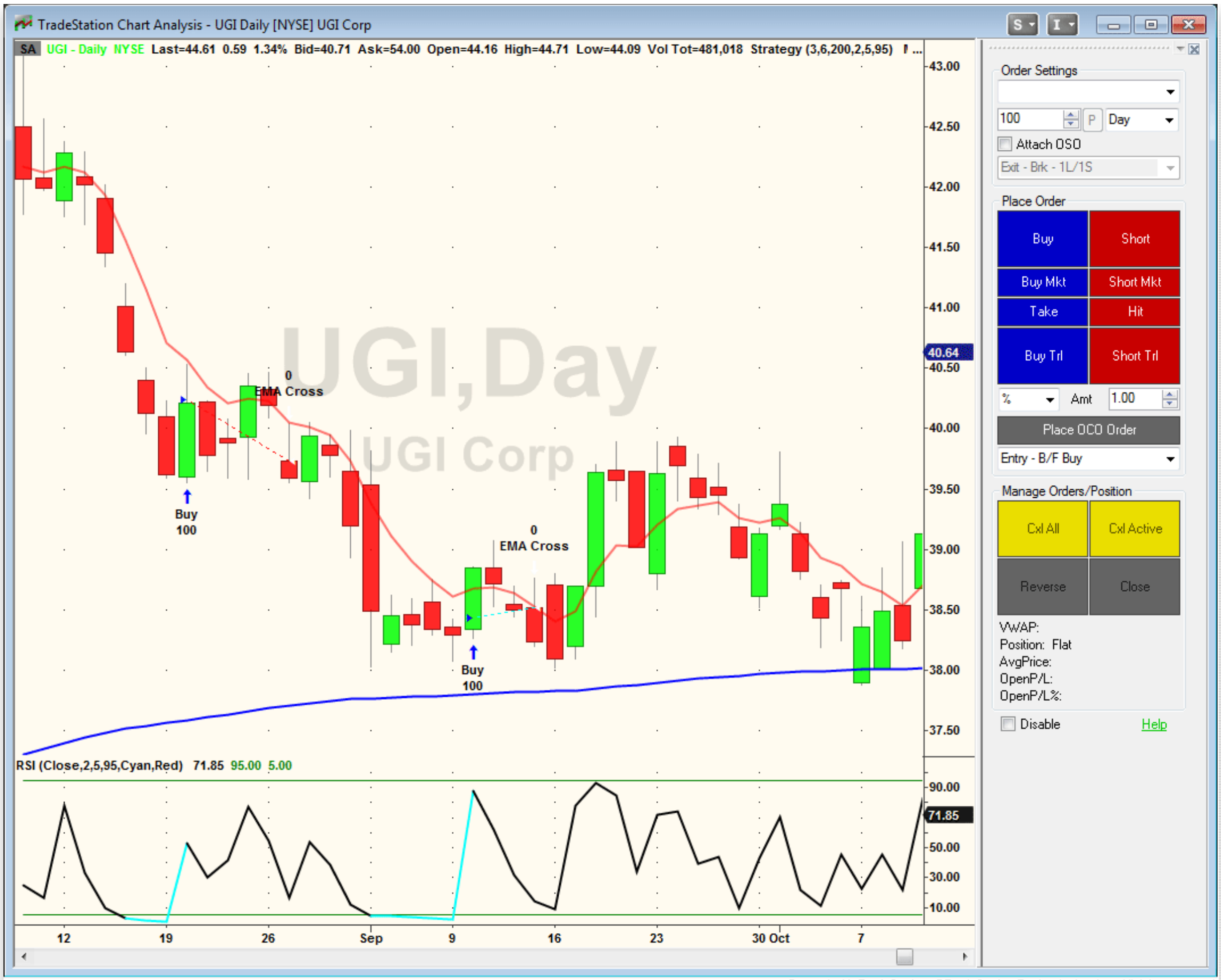
We are providing the strategy code for the author's entry and exit rules. In addition to backtesting the strategy in a TradeStation chart, remember that you can use TradeStation's Portfolio Maestro product to quickly backtest on a portfolio of symbols of your choice.

To download the EasyLanguage code, please visit our TradeStation and EasyLanguage support forum. The code for this technique can be found here: http://www.tradestation.com/TASC-2014. The ELD filename is "_TASC_LongHaul.ELD." (The code is also shown below.)

```
inputs:
        NumBarsForTrail( 3 ),
        EMALength( 6 ),
        EMAFilterLength( 200 ),
        RSILength( 2 ),
        RSIOverSoldLevel( 5 ),
        RSIOverBoughtLevel( 95 ) ;

variables:
        RSIValue( 0 ),
        EMAFastValue( 0 ),
        EMAFilterOK( false ),
        ReversalOK( false ) ,
        RSISetUpOK( false ),
        MP( 0 ) ;

MP = MarketPosition ;
RSIValue = RSI( Close, RSILength ) ;
EMAFastValue = XAverage( Close, EMALength ) ;

EMAFilterOK = Close >
        XAverage( Close, EMAFilterLength ) ;
ReversalOK = High < High[1]
        and Close < Open and Close[1] < Open[1] ;

if MP = 0 and RSIValue < RSIOverSoldLevel then
        RSISetUpOK = true
else if MP = 1 or RSIValue > RSIOverBoughtLevel then
        RSISetUpOK = false ;


if RSISetUpOK and ReversalOK and EMAFilterOK then
        Buy next bar at High + MinMove / PriceScale Stop ;

if MP = 1 then
begin
        Sell ( "3B Trail" )next bar
                at Lowest( Low[1], 3 ) Stop ;
        if Close crosses under EMAFastValue then
                Sell ( "EMA Cross" )next bar at Market ;
        end ;
```

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

A sample chart is shown in Figure 1.



Created with TradeStation. ©TradeStation Technologies, Inc. All rights reserved.

FIGURE 1: TRADESTATION. This chart shows Donald Pendergast's strategy applied to a daily chart of UGI along with the six- and 200-period exponential moving averages and two-period RSI indicators.

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

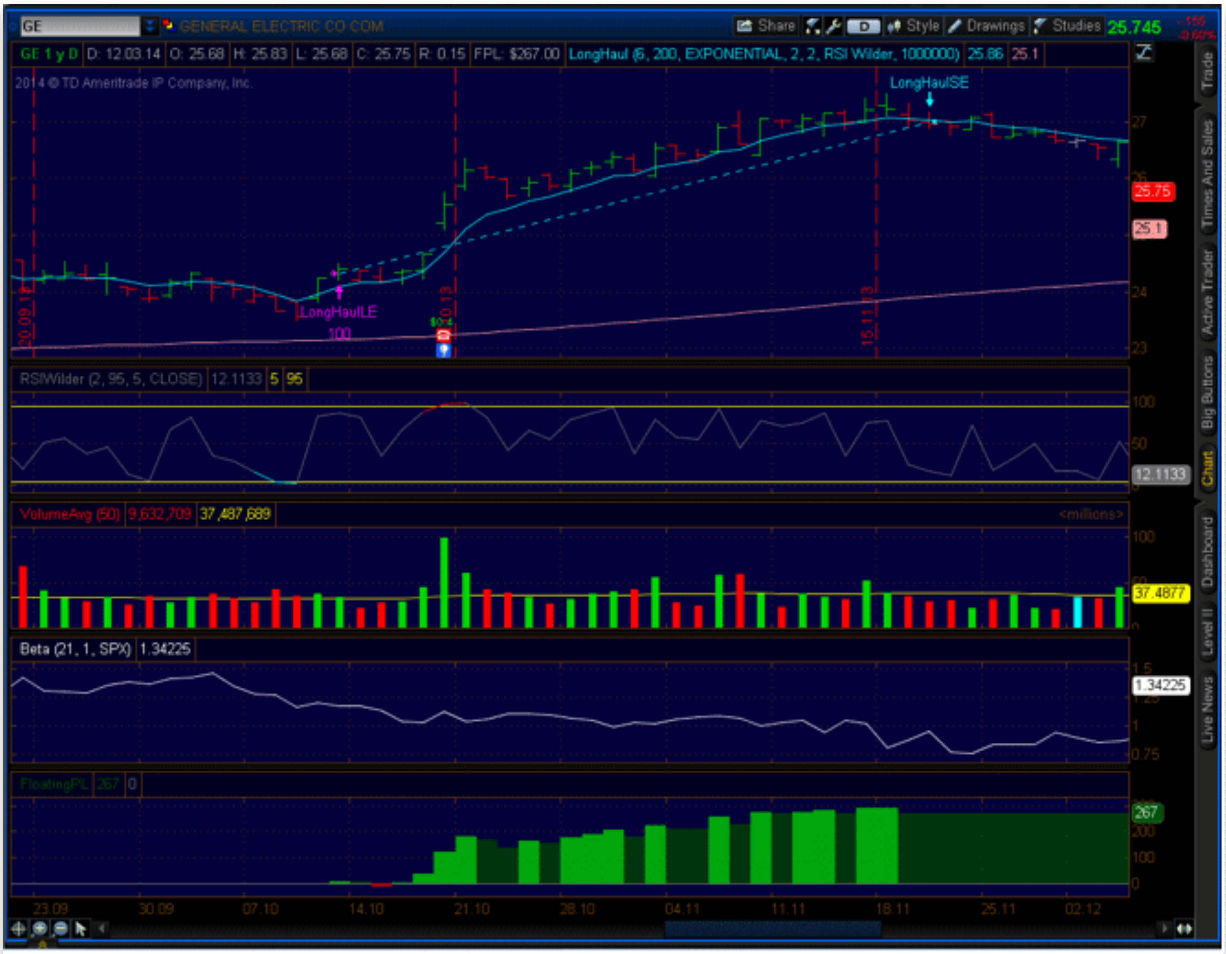
BACK TO LIST

THINKORSWIM: MAY 2014

In “A Trading Method For The Long Haul,” which appeared in the 2014 Bonus Issue of STOCKS & COMMODITIES, author Donald W. Pendergast Jr. serves up a model that is “dirt-simple,” per his own words. One of the hardest parts of trading is staying disciplined. That is why a simple trading model like the one described in Pendergast’s article is perfect for many investors.

At thinkorswim, we make the loading process easy. Simply click on the link http://tos.mx/flH5JZ and choose *backtest*. You can adjust the parameters within the *edit studies* window to fine-tune your variables.

The chart in Figure 2 shows the entry and exit point when the criteria described in the article was met (based on the RSI, volume average, and beta.) In thinkorswim, backtesting is done with a tool called *strategies*. The link mentioned earlier will save the strategy for you. Then you will have the ability to load the strategy and begin backtesting.


GE 1 y D | D: 12.03.14 | O: 25.68 | H: 25.83 | L: 25.68 | C: 25.75 | R: 0.15 | FPL: $267.00 | LongHaul (6, 200, EXPONENTIAL, 2, 2, RSI Wilder, 1000000) | 25.86 | 25.1

RSIWilder (2, 95, 5, CLOSE) | 12.1133 | 5 | 95

VolumeAvg (50) | 9,632,709 | 37,487,689

Beta (21, 1, SPX) | 1.34225

FloatingPL | 267 | 0

FIGURE 2: THINKORSWIM. The chart shows the entry and exit point when the criteria described in Donald Pendergast’s article in the 2014 Bonus Issue was met (based on RSI, volume average, and beta.)

Pendergast’s article also discusses scanning for stocks that meet his defined criteria. Don’t worry — thinkorswim has you covered there as well. Just click the following link to load his predefined scan: http://tos.mx/WE2FzG. Now you can backtest your known stocks as well as find new ones that meet this model’s criteria, all with a couple of clicks. Happy swimming!

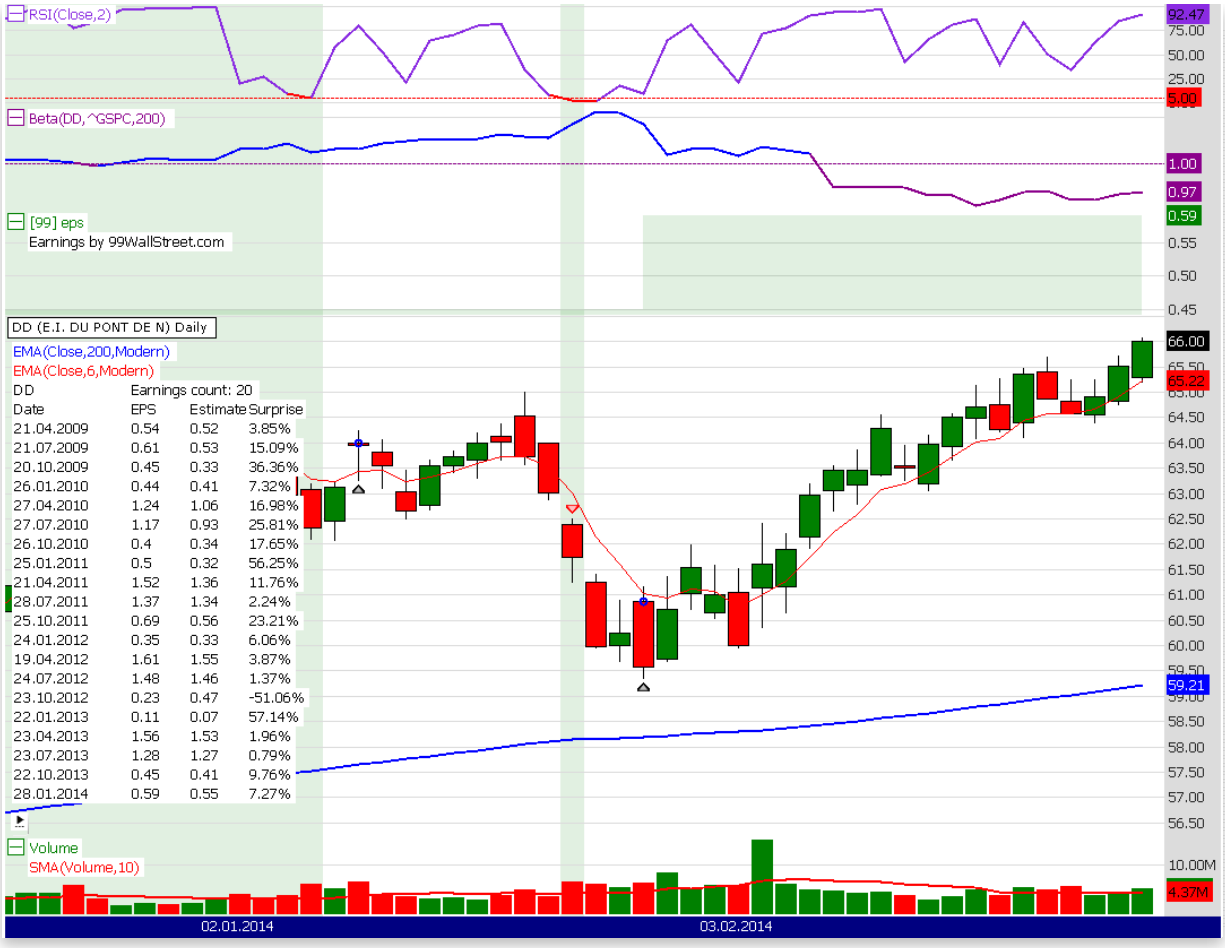
—thinkorswim
A division of TD Ameritrade, Inc.
www.thinkorswim.com

BACK TO LIST

WEALTH-LAB: MAY 2014

It’s usually interesting to see if there’s added benefit from including non-price information such as fundamental items or sentiment data into a system’s rules. In addition to having an intermarket filter (beta), the system featured in Donald W. Pendergast’s article that appeared in the 2014 Bonus Issue of Stocks & Commodities (“A Trading Method For The Long Haul”) puts emphasis on beating a stock’s most recent quarterly earnings estimates as one of the key elements. To be able to backtest its rules and generate signals for live trading, Wealth-Lab customers should use one of the fundamental data sources that provides access to earnings-surprise data, for example, 99WallStreet.com, for which a fundamental provider already exists.

The strategy we’re using to illustrate the trading method adheres to Pendergast’s bullet-point list of entry criteria with one exception: It’s not required for a stock to have a strong history of positive earnings surprises over the last couple of years, as that would limit the number of signals to a mere handful. To enter, the system would scout for stocks where the five technical, intermarket, and fundamental conditions are in place, and then wait while a stock’s two-day RSI crosses above 5 from oversold territory. As we found the proposed exit strategy triggering too many false signals, not allowing the system to operate to its full potential, we replaced it with a simple trailing exit at a 20-day stop of the daily lows (Figure 3).



DD Earnings count: 20

Date	EPS	Estimate	Surprise
21.04.2009	0.54	0.52	3.85%
21.07.2009	0.61	0.53	15.09%
20.10.2009	0.45	0.33	36.36%
26.01.2010	0.44	0.41	7.32%
27.04.2010	1.24	1.06	16.98%
27.07.2010	1.17	0.93	25.81%
26.10.2010	0.4	0.34	17.65%
25.01.2011	0.5	0.32	56.25%
21.04.2011	1.52	1.36	11.76%
28.07.2011	1.37	1.34	2.24%
25.10.2011	0.69	0.56	23.21%
24.01.2012	0.35	0.33	6.06%
19.04.2012	1.61	1.55	3.87%
24.07.2012	1.48	1.46	1.37%
23.10.2012	0.23	0.47	-51.06%
22.01.2013	0.11	0.07	57.14%
23.04.2013	1.56	1.53	1.96%
23.07.2013	1.28	1.27	0.79%
22.10.2013	0.45	0.41	9.76%
28.01.2014	0.59	0.55	7.27%

FIGURE 3: WEALTH-LAB. This Wealth-Lab 6 chart shows the application of the fundamental and technical rules from Donald Pendergast's article in the 2014 Bonus Issue on a daily chart of DD (Dupont). The upper pane shows the two-day RSI, below that is the 200-day beta indicator, on top of price there are historical quarterly earnings releases, and volume with average 10-day volume are at the bottom.

There are two prerequisites to successfully execute the included strategy in Wealth-Lab:

- Install the Community Indicators library to build the beta indicator
- Install 99WallStreet fundamental data provider *and* download the earnings surprise data for your DataSet.

Latest versions of both libraries are available to registered Wealth-Lab customers for download from the *extensions* section of our website (restart Wealth-Lab after installation or update).

The code listing is also shown below.

C# Code

```
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 TASC201405Pendergast : WealthScript
        {
                protected override void Execute()
                {
                        const char tab =
'\u0009';
                        string item = "[99] eps";
                        IList<FundamentalItem> fList = FundamentalDataItems(item);
                        DataSeries e = FundamentalDataSeries(item);
                        ChartPane ep = CreatePane( 20, true, true );
                        PlotFundamentalItems(ep, item, Color.Green,
WealthLab.LineStyle.Invisible, 1);
                        DrawLabel(ep, "Earnings by 99WallStreet.com");

                        if( fList.Count == 0 )
                        {
                                DrawLabel( PricePane, "Update 99WallStreet.com data
before running the Strategy");
                                Abort();
                        }

                        DrawLabel(PricePane, Bars.Symbol + tab + tab + "Earnings
count: " + fList.Count);
                        DrawLabel(PricePane, "Date" + tab + tab + "EPS" + tab +
"Estimate" + tab + "Surprise");
                        foreach (FundamentalItem fi in fList)
                        {
                                DrawLabel(PricePane,
                                        fi.Date.ToShortDateString() + tab +
                                        fi.Value.ToString() + tab +
                                        fi.GetDetail("EPSEstimate") + tab +
                                        fi.GetDetail("EPSSurprise"));
                        }

                        Bars spx = GetExternalSymbol("^GSPC", true); // Yahoo data
                        //Bars spx = GetExternalSymbol(".SPX", true); // Fidelity data
                        Beta beta = Beta.Series( Bars, spx,200);
                        ChartPane pb = CreatePane( 20, true, true );
                        PlotSeries( pb, beta, Color.DarkMagenta, LineStyle.Solid, 2 );
                        DrawHorzLine( pb, 1.0, Color.DarkMagenta, LineStyle.Dashed, 1
);

                        DataSeries smaAvgVol = SMA.Series(Volume, 10);
                        PlotSeries( VolumePane, smaAvgVol, Color.Red,
LineStyle.Solid, 2 );

                        DataSeries ema = EMA.Series(Close,200,EMACalculation.Modern);
                        DataSeries ema6 = EMA.Series( Close,6,EMACalculation.Modern);
                        PlotSeries( PricePane, ema, Color.Blue, LineStyle.Solid, 2 );
                        PlotSeries( PricePane, ema6, Color.Red, LineStyle.Solid, 1 );
                        DataSeries rsi = RSI.Series(Close,2);
                        ChartPane pr = CreatePane( 20, true, true );
                        PlotSeries( pr, rsi, Color.BlueViolet, LineStyle.Solid, 2 );
                        DrawHorzLine( pr, 5, Color.Red, LineStyle.Dashed, 1 );
                        for(int bar = 2; bar < Bars.Count; bar++)
                        {
                                if ( rsi[bar] < 5 )
                                        SetSeriesBarColor( bar, rsi, Color.Red );
                                if ( beta[bar] > 1 )
                                        SetSeriesBarColor( bar, beta, Color.Blue );
                        }

                        bool SetupValid = false;

                        for(int bar = GetTradingLoopStartBar(200); bar < Bars.Count;
bar++)
                        {
                                if (IsLastPositionActive)
                                {
                                        SellAtTrailingStop( bar+1, LastPosition,
Lowest.Series( Low, 20 )[bar], "Trailing Stop" );
                                }
                                else
                                {
                                        if( !SetupValid )
                                        {
                                                bool rsi2 = rsi[bar] < 5;
                                                bool ema200 = Close[bar] > ema[bar];
                                                bool avgVol = smaAvgVol[bar] >=
1000000;

                                                bool surprise = false;

                                                for(int i = fList.Count-1; i >= 0;
i--)

                                                {
                                                        if( bar > fList[i].Bar )
                                                        {
                                                                surprise = (
Double.Parse( fList[i].GetDetail("EPSSurprise").TrimEnd(new char[] {'%'} )) > 0 );
                                                                break;
                                                        }
                                                }

                                                bool bet = beta[bar] > 1.0;

                                                if( surprise )
                                                        SetBackgroundColor( bar,
Color.FromArgb(30,Color.Green) );

                                                if( rsi2 && ema200 && avgVol &&
surprise && bet )
                                                        SetupValid = true;
                                        }

                                        if( SetupValid )
                                        {
                                                if( CrossOver( bar, rsi, 5 ) )
                                                        if( BuyAtMarket(bar+1) !=
null )
                                                                SetupValid = false;
```

```
                    }
                }
            }
        }
    }
}
```

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

BACK TO LIST

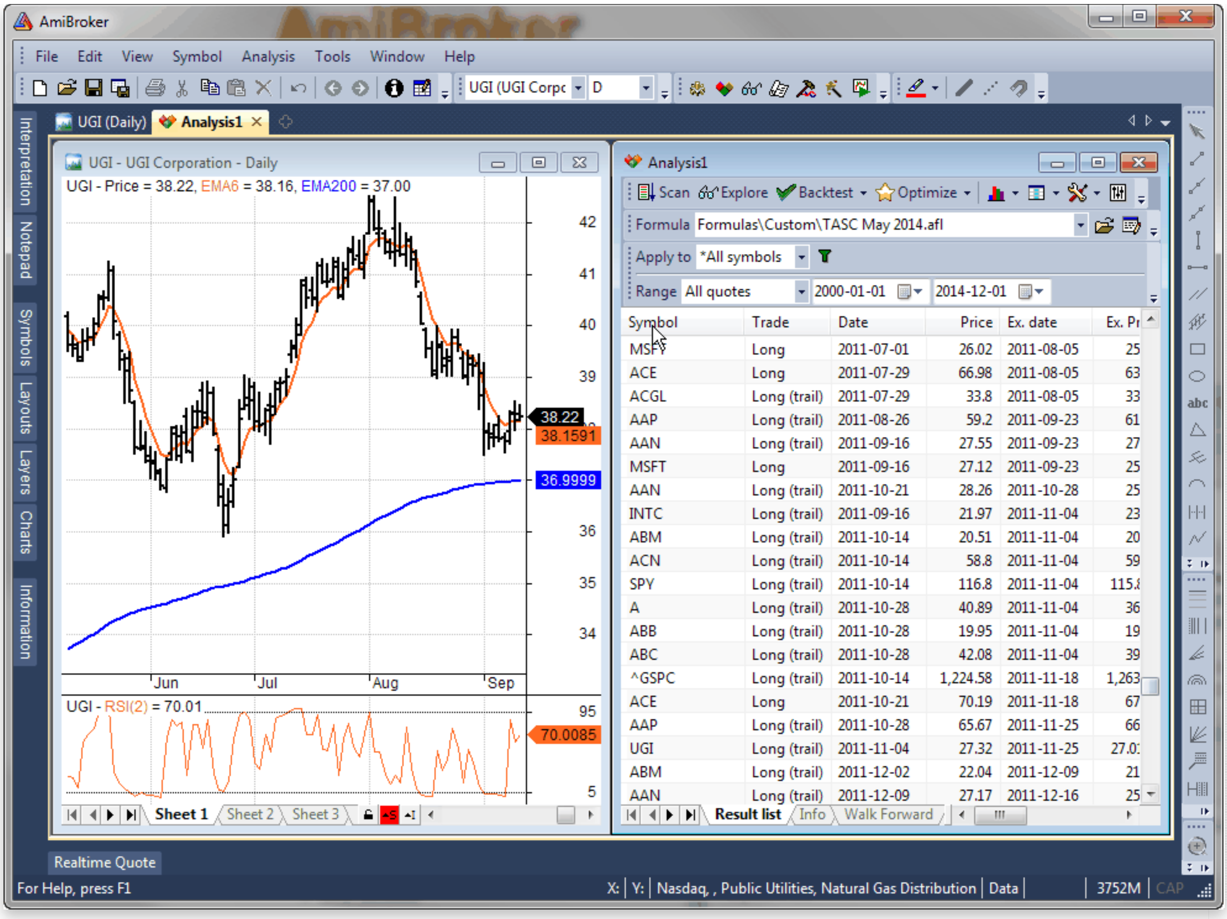
AMIBROKER: MAY 2014

In “A Trading Method For The Long Haul,” which appeared in the 2014 Bonus Issue of STOCKS & COMMODITIES, author Donald W. Pendergast Jr. presents a long-only strategy based on the RSI, channel breakouts, and moving averages.

Our ready-to-use formula based on Pendergast’s article has three purposes: It works as a screening tool (called an *exploration* in AmiBroker) to filter out trade candidates meeting the criteria described in the article. It also can be used as a set of trading rules to backtest the system, and it can also be used as a chart.

To use the formula, copy it to the AFL formula editor, then press “send to analysis” button. From the *analysis* window, you can press the *explore* button to run a screening and/or press *backtest* to evaluate trading system performance.

The code listing is shown below. A sample chart is shown in Figure 4.



Formula Formulas\Custom\TASC May 2014.afl

Apply to *All symbols

Range All quotes 2000-01-01 2014-12-01

Symbol	Trade	Date	Price	Ex. date	Ex. Pr
MSFT	Long	2011-07-01	26.02	2011-08-05	25
ACE	Long	2011-07-29	66.98	2011-08-05	63
ACGL	Long (trail)	2011-07-29	33.8	2011-08-05	33
AAP	Long (trail)	2011-08-26	59.2	2011-09-23	61
AAN	Long (trail)	2011-09-16	27.55	2011-09-23	27
MSFT	Long	2011-09-16	27.12	2011-09-23	25
AAN	Long (trail)	2011-10-21	28.26	2011-10-28	25
INTC	Long (trail)	2011-09-16	21.97	2011-11-04	23
ABM	Long (trail)	2011-10-14	20.51	2011-11-04	20
ACN	Long (trail)	2011-10-14	58.8	2011-11-04	59
SPY	Long (trail)	2011-10-14	116.8	2011-11-04	115.8
A	Long (trail)	2011-10-28	40.89	2011-11-04	36
ABB	Long (trail)	2011-10-28	19.95	2011-11-04	19
ABC	Long (trail)	2011-10-28	42.08	2011-11-04	39
^GSPC	Long (trail)	2011-10-14	1,224.58	2011-11-18	1,263
ACE	Long	2011-10-21	70.19	2011-11-18	67
AAP	Long (trail)	2011-10-28	65.67	2011-11-25	66
UGI	Long (trail)	2011-11-04	27.32	2011-11-25	27.0
ABM	Long (trail)	2011-12-02	22.04	2011-12-09	21
AAN	Long (trail)	2011-12-09	27.17	2011-12-16	25

Result list / Info / Walk Forward

FIGURE 4: AMIBROKER. Here is a daily chart of UGI with six-day and 200-day exponential moving averages, a two-day RSI (left side), and an analysis trade list (right).

```
// TASC May 2014 - A Trading Method For The Long Haul
// - article by Donald W. Pendergast Jr.
//
// Screening part / Exploration
//

RSI2 = RSI( 2 );
EMA200 = EMA( C, 200 );
AvgVol = MA( V, 50 );

SP = Foreign( "SPY", "C" );

RelPerf52Week = ROC( C / SP, 252 );
RelPerf26Week = ROC( C / SP, 126 );
RelPerf13Week = ROC( C / SP, 61 );
RelPerf4Week = ROC( C, 20 );
StartYear = Year() != Ref( Year(), -1 );
SYV = ValueWhen( StartYear, C / SP );
RelPerfYTD = 100 * ( ( C / SP ) - SYV ) / SYV;

Filter = RSI2 < 5 AND C > EMA200 AND AvgVol > 1000000;

AddColumn( RSI2, "RSI2" );
AddColumn( EMA200, "EMA200" );
AddColumn( AvgVol, "AvgVol" );
AddColumn( RelPerf4Week, "RelPerf4Week" );
AddColumn( RelPerf13Week, "RelPerf13Week" );
AddColumn( RelPerf26Week, "RelPerf26Week" );
AddColumn( RelPerf52Week, "RelPerf52Week" );
AddColumn( RelPerfYTD , "RelPerfYTD" );


// trading rules part
SetPositionSize( 10, spsPercentOfEquity ); // 10% in single trade
SetOption("MaxOpenPositions", 10 ); // 10 positions max.

Buy = Hold( Filter, 40 ) // screening criteria met within last 40 bars
      AND
      Close > Ref( HHV( High, 5 ), -1 ); // stock closes above highs of last 5 bars


Sell = Cross( EMA( C, 6 ), C ); // exit when it drops below 6 period EMA

ApplyStop( stopTypeTrailing, stopModePoint, BuyPrice - Ref( LLV( Low, 3 ), -1 ),
True, True ); // 3-bar LLV trailing stop


// indicator part
Plot( C, "Price", colorDefault, styleBar | styleThick );
Plot( EMA( C, 6 ), "EMA6", colorOrange, styleThick );
Plot( EMA200, "EMA200", colorBlue, styleThick  );
```

—Tomasz Janeczko
AmiBroker.com
www.amibroker.com

BACK TO LIST

NEUROSHELL TRADER: MAY 2014

A trigger trading system based on the system described by Donald W. Pendergast Jr. in his 2014 Bonus Issue article in Stocks & Commodities, “A Trading Method For The Long Haul,” can be easily implemented in NeuroShell Trader with a few of the program’s 800+ indicators. Simply select “New trading strategy” from the Insert menu and enter the following formulas in the appropriate locations of the trading strategy wizard:

```
BUY LONG CONDITIONS: [All of which must be true]
A<B(Min(RSI(Close,2),25),5)
     CrossAbove(Close,Lag(Max(High,25),1))


LONG TRAILING STOP PRICES:
```

```
IfThenElse(A<B(Close,ExpAvg(Close,6)),MaxValEntryAct(Trading Strategy,Min(Low,3),1),*)
```

```
SELL LONG CONDITIONS
CrossBelow(Close,ExpAvg(Close,6))
```

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 website to download a copy of this or any previous Traders' Tips.

A sample chart is shown in Figure 5.

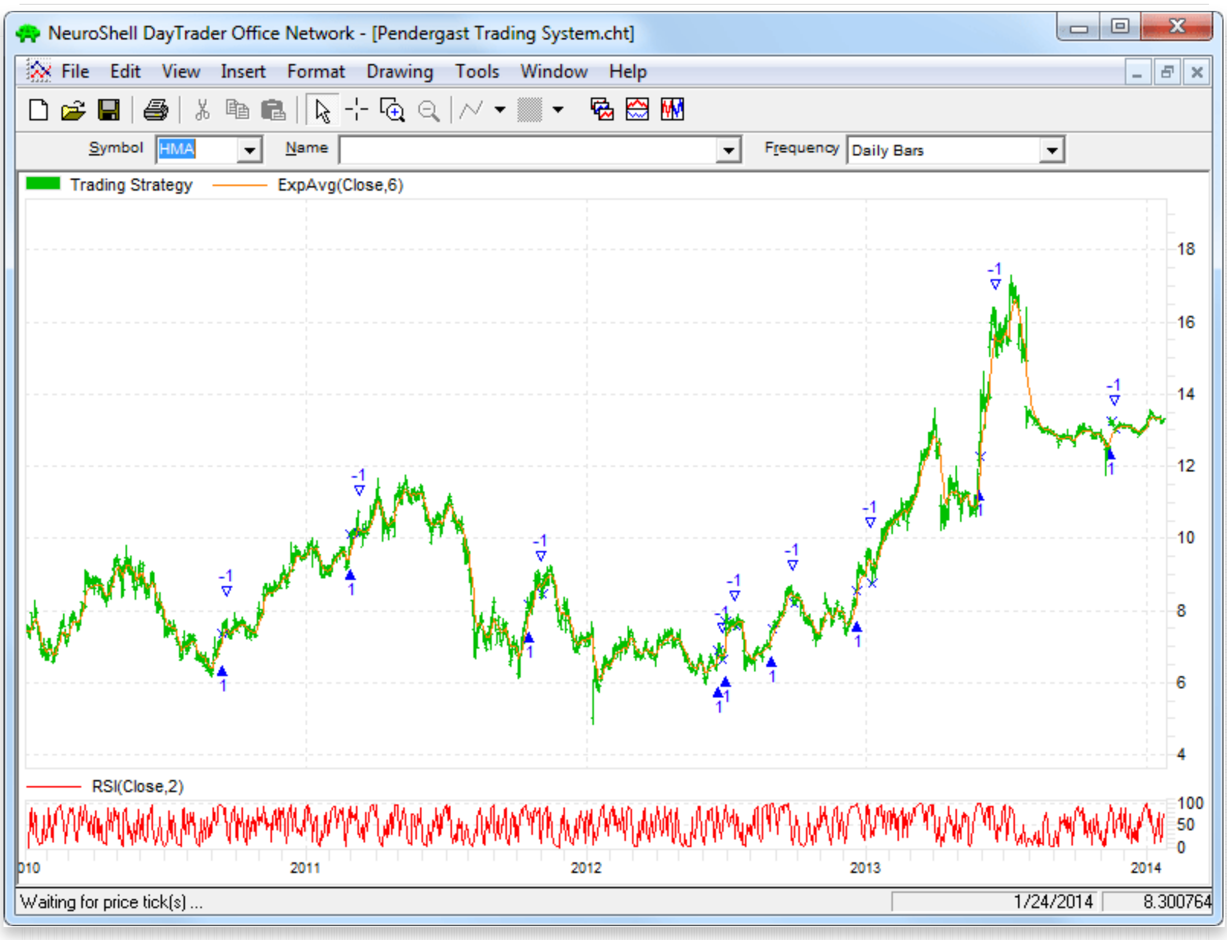


FIGURE 5: NEUROSHELL TRADER. This NeuroShell Trader chart displays the trading system for HMA.

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

BACK TO LIST

AIQ: MAY 2014

The AIQ code and EDS file based on Donald W. Pendergast's article in the 2014 Bonus Issue of STOCKS & COMMODITIES, "A Trading Method For The Long Haul," can be found at www.TradersEdgeSystems.com/traderstips.htm.

The code I provide there for the long haul system is modified somewhat from the author's descriptions as follows. First, I did not implement the fundamental rule, but this can be done if a data source is located that can export the fundamental fields needed for each stock into a .csv file. This could then be imported into the fundamental module. Second, I modified the exit to add an RSI profit target and changed some of the exit parameters.

To get the code to run properly, the AIQALL list of stocks and groups must be installed and updated on the user's computer. To do this, first get the most recent AIQALL list from the AIQ website, then add all the stocks from the latest data disk that have trading volume greater than about 200,000 shares. We need these in order to have enough stocks to compute the group indexes. Next, we would download data for all the stocks in the database up to the current date. Then, as shown in Figure 6, we would set the RS tickers to the AIQALL list, and also, as shown in Figure 7, recompute all dates for all the groups in the AIQALL list.

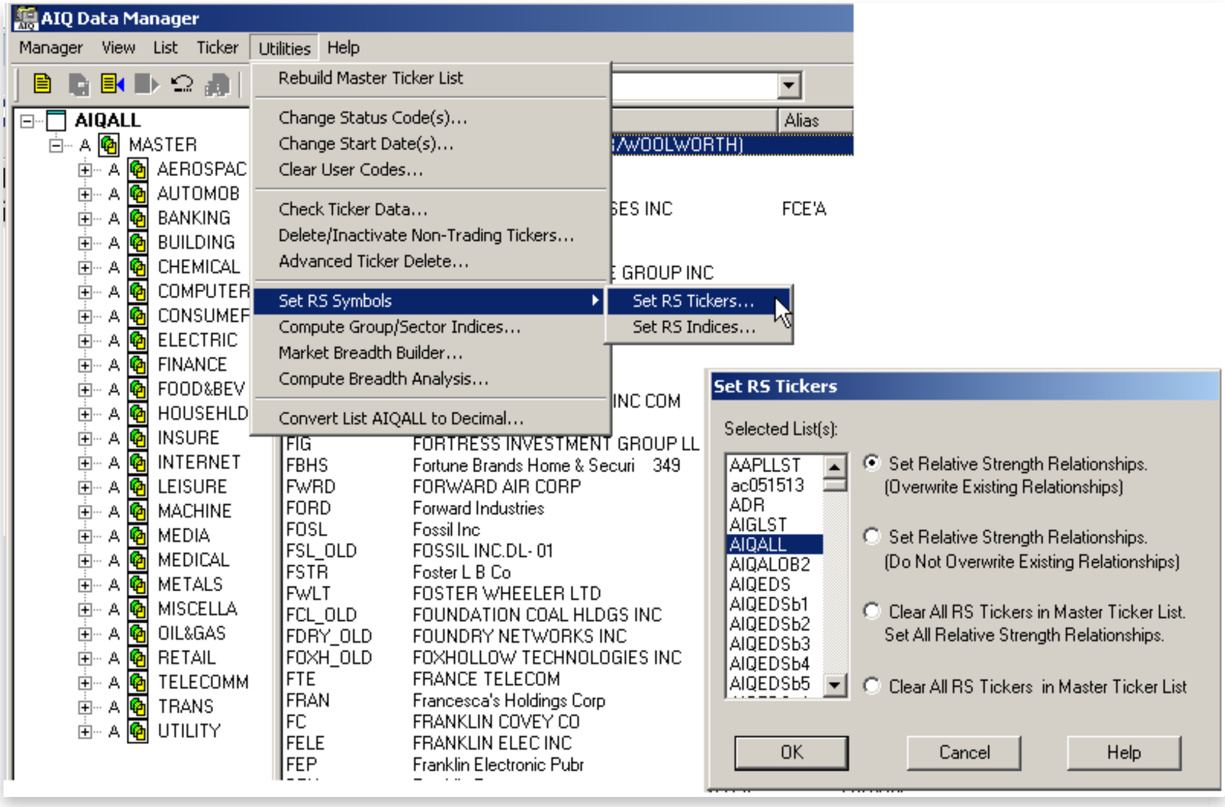
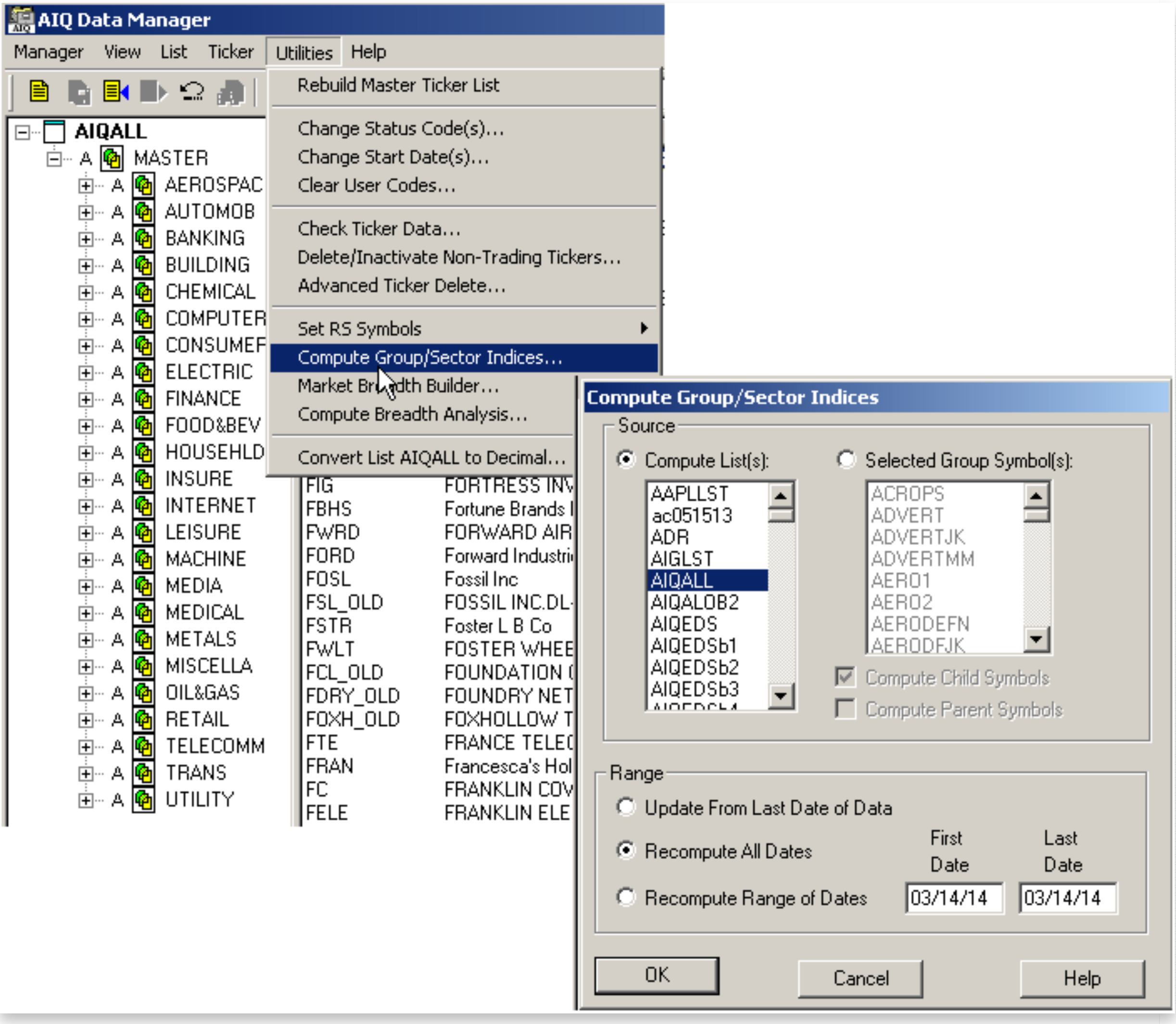

AIQ Data Manager
Manager View List Ticker Utilities Help
Rebuild Master Ticker List
Change Status Code(s)...
Change Start Date(s)...
Clear User Codes...
Check Ticker Data...
Delete/Inactivate Non-Trading Tickers...
Advanced Ticker Delete...
Set RS Symbols
Compute Group/Sector Indices...
Market Breadth Builder...
Compute Breadth Analysis...
Convert List AIQALL to Decimal...
Set RS Tickers...
Set RS Indices...
Alias
AIQALL
A MASTER
A AEROSPAC
A AUTOMOB
A BANKING
A BUILDING
A CHEMICAL
A COMPUTER
A CONSUMEF
A ELECTRIC
A FINANCE
A FOOD&BEV
A HOUSEHLD
A INSURE
A INTERNET
A LEISURE
A MACHINE
A MEDIA
A MEDICAL
A METALS
A MISCELLA
A OIL&GAS
A RETAIL
A TELECOMM
A TRANS
A UTILITY
/WOOLWORTH)
ES INC FCE'A
GROUP INC
INC COM
FIG FORTRESS INVESTMENT GROUP LL
FBHS Fortune Brands Home & Securi 349
FWRD FORWARD AIR CORP
FORD Forward Industries
FOSL Fossil Inc
FSL_OLD FOSSIL INC.DL-01
FSTR Foster L B Co
FWLT FOSTER WHEELER LTD
FCL_OLD FOUNDATION COAL HLDGS INC
FDRY_OLD FOUNDRY NETWORKS INC
FOXH_OLD FOXHOLLOW TECHNOLOGIES INC
FTE FRANCE TELECOM
FRAN Francesca's Holdings Corp
FC FRANKLIN COVEY CO
FELE FRANKLIN ELEC INC
FEP Franklin Electronic Pubr
Set RS Tickers
Selected List(s):
AAPLLST
ac051513
ADR
AIGLST
AIQALL
AIQALOB2
AIQEDS
AIQEDSb1
AIQEDSb2
AIQEDSb3
AIQEDSb4
AIQEDSb5
● Set Relative Strength Relationships. (Overwrite Existing Relationships)
○ Set Relative Strength Relationships. (Do Not Overwrite Existing Relationships)
○ Clear All RS Tickers in Master Ticker List. Set All Relative Strength Relationships.
○ Clear All RS Tickers in Master Ticker List
OK
Cancel
Help

FIGURE 6: AIQ DATA MANAGER. Use the AIQ Data Manager to set the RS tickers to the AIQALL list.



● Compute List(s): AIQALL

○ Selected Group Symbol(s):

☑ Compute Child Symbols

□ Compute Parent Symbols

○ Update From Last Date of Data

● Recompute All Dates

○ Recompute Range of Dates 03/14/14 03/14/14

FIGURE 7: AIQ DATA MANAGER. Use the AIQ Data Manager to compute the group & sector indexes for the AIQALL list.

The EDS file containing the code has the properties set to the AIQALL list. If you are building an EDS file directly from the code listing below, then be sure to set the properties to the AIQALL list.

```
!A Trading Method for the Long Haul
!Author: Donald W. Pendergast Jr., TASC Bonus Issue 2014
!Coded by: Richard Denning 3/10/14
!www.TradersEdgeSystems.com

!INPUTS:
trendLen is 200.
rsiLen is 2.
minAvgVolume is 10000.
volaLen is 21.
```

```
relStrLen is 80.
minVolaRatio is 0.5.
rsSPXmin is 1.0.
rsGroupmin is 1.0.
trailBars is 2.
rsiBuyLvl is 5.
rsiExitLvl is 95.
exitLen is 18.
C is [close].
H is [high].
L is [low].
PD is {position days}.

!LONG TERM MOVING AVERAGE:
emaLT is expavg(C,trendLen).
emaST is expavg(L,exitLen).

!RSI WILDER
!To convert Wilder Averaging to Exponential Averaging use this formula:
!ExponentialPeriods = 2 * WilderPeriod - 1.
U is C-valresult(C,1).
D is valresult(C,1)-C.
rsiLen1 is 2 * rsiLen - 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))).

!VOLATILITY
price1 is H.
price2 is L.
ratio is price1 / price2.
dp is Ln(ratio).
dpsqr is Ln(ratio) * Ln(ratio).
totdpsqr is sum(dpsqr,volaLen).
sumdp is sum(dp,volaLen).
sumdpsqr is sumdp * sumdp.
sumdpave is sumdpsqr / volaLen.
diff is totdpsqr - sumdpave.
!!use 252 for daily, or 52 for weekly below
factor is 252 / (volaLen-1).
result is sqrt(diff * factor).
vola  is result * 100 .
volaAvg is expavg(vola,volaLen).
volaSPXavg is tickerUDF("SPX",volaAvg).
volaRatio is volaSPXavg/volaAvg.

!AVERAGE VOLUME
avgVolume is expavg([volume],50).

!RELATIVE STRENGTH
roc is C / valresult(C,relStrLen).
rocSPX is tickerUDF("SPX",roc).
rocGroup is tickerUDF(rsticker(),roc).
groupSymbol is tickerUDF(rsticker(),symbol()).
groupName is tickerUDF(rsticker(),description()).
rsSPX is roc / rocSPX.
rsGroup is rocGroup / rocSPX.

!SCREENING RULES
VolumeRule if avgVolume > minAvgVolume.
```

```
TrendRule if C > emaLT.
VolaRule if volaRatio > minVolaRatio and simpleavg(H-L,10) > simpleavg(H-L,200).
RelStrRule if rsSPX > rsSPXmin and rsGroup > rsGroupmin.
PullbackRule if rsi < rsiBuyLvl.
EnoughData if hasdatafor(trendLen+10) > trendLen.
!FundamentalRule if [eps]>[eps est].

Screen if EnoughData
        and TrendRule
        and VolaRule
        and RelStrRule
        and PullbackRule
        !and FundamentalRule
        and VolumeRule.

EntryTrigger if C > lowresult(H,2,1).
Buy if valrule(Screen,1) and EntryTrigger.

Exit if (C < lowresult(L,trailBars,1))
        or
        (valrule(C > emaST,1) and C < emaST)
        or
        rsi > rsiExitLvl.

ShowValues if EnoughData.
```

—Richard Denning
info@TradersEdgeSystems.com
for AIQ Systems

BACK TO LIST

TRADERSSTUDIO: MAY 2014

The TradersStudio code based on Donald W. Pendergast's article in the 2014 Bonus Issue of STOCKS & COMMODITIES, issue, "A Trading Method For The Long Haul," 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:

- System LONG_HAUL: A trading system that employs most of the rules outlined in Pendergast's 2014 Bonus Issue article, "A Trading Method For The Long Haul,"
- Function VOLATILITY: Returns the volatility value based on a standard deviation-type formula
- Function SUMM: Returns the summation value of the price array input (helper function for

volatility).

To set up the system as Pendergast outlines, we would create a session for each group of stocks that will be traded. *Independent1* should always be set to a major market index like the S&P 500, and *independent2* should be set to the group index for the stocks that are included in the session. Once a session is set up for each group to be traded, a trade plan could be written that would control how many trades are accepted from each group as well as the size of each trade. I did not write the trade plan due to time constraints, but I did test the system on just one group, the NASDAQ 100 group of stocks. I set *independent1* to the S&P 500 index (SPX) and *independent2* to the NASDAQ 100 index (NDX). I did a minor amount of optimization and also added an RSI profit target exit. My code does not implement the fundamental rule.

In Figure 8, I show the equity curve and underwater equity curve for the single session trading a fixed 200 shares of each stock.

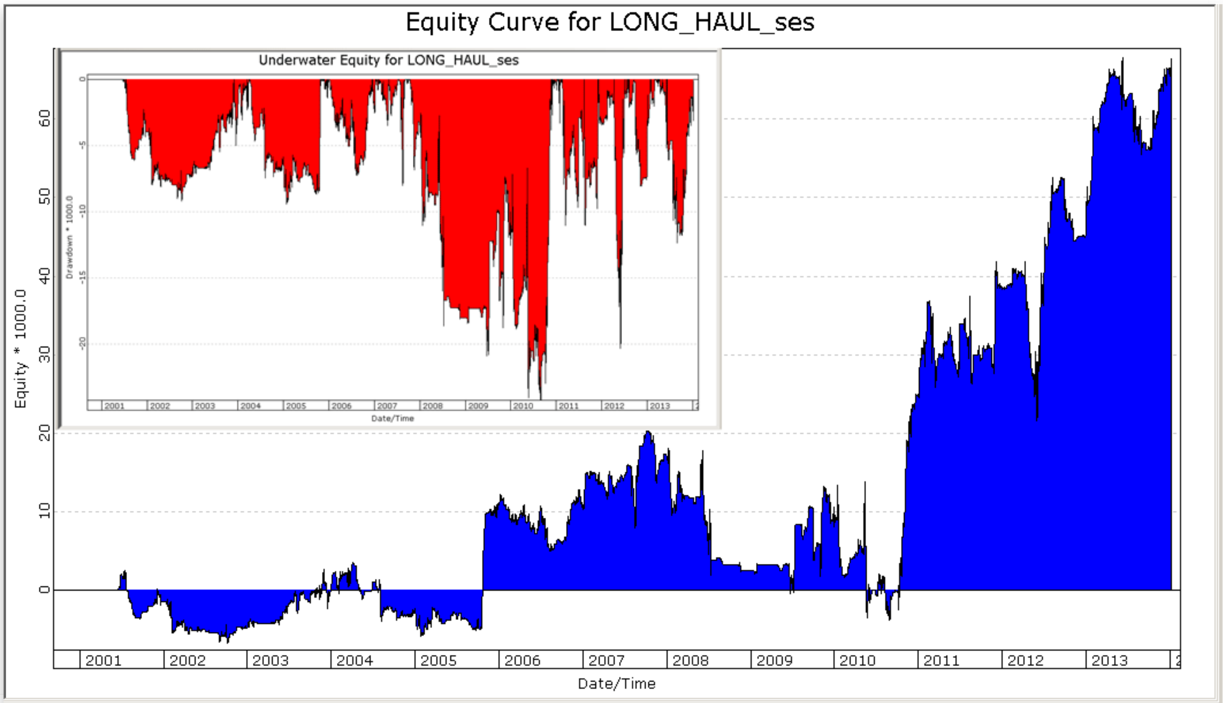


FIGURE 8: TRADERSSTUDIO. Here is an equity curve (blue) and underwater equity curve (red) for Donald Pendergast's long haul system trading a single group, the NASDAQ 100 group of stocks.

```
'A Trading Method for the Long Haul
'Author: Donald W  Pendergast Jr , TASC Bonus Issue 2014
'Coded by: Richard Denning 3/10/14
'www TradersEdgeSystems com

'LONG HAUL TRADING SYSTEM:
Sub
LONG_HAUL(trendLen,rsiLen,minAvgVolume,volaLen,relStrLen,minVolaRatio,rsSPXmin,rsGroup
min,trailBars,rsiBuyLvl,rsiExitLvl,exitLen)
```

```
'LONG & SHORT TERM MOVING AVERAGES:
Dim emaLT As BarArray
Dim emaST As BarArray
emaLT  = XAverage(C,trendLen)
emaST  = XAverage(L,exitLen)

'RSI WILDER
Dim rsiWilder As BarArray
rsiWilder = rsi(C,rsiLen,0)

'VOLATILITY
Dim ratio As BarArray
Dim ratioINDX As BarArray
Dim volaAvg As BarArray
Dim volaAvgINDX As BarArray
Dim IndexC As BarArray
Dim IndexH As BarArray
Dim IndexL As BarArray
Dim GroupC As BarArray
IndexC = C Of independent1
IndexH = H Of independent1
IndexL = L Of independent1
GroupC = C Of independent2
If H - L <> 0 then ratio  = (H - L )/((H + L)/2)
If IndexL>0 Then ratioINDX = IndexH / IndexL
volaAvg = Average(volatility(ratio,volaLen),volaLen)
volaAvgINDX = Average(volatility(ratioINDX,volaLen),volaLen)

'AVERAGE VOLUME
Dim avgVolume As BarArray
avgVolume  = XAverage(V,50)

'RELATIVE STRENGTH
Dim rocStk As BarArray
Dim rocINDX As BarArray
Dim rocGroup As BarArray
If C[relStrLen]>0 Then rocStk  = C / C[relStrLen]
If IndexC[relStrLen]>0 Then rocINDX  = IndexC / IndexC[relStrLen]
If GroupC[relStrLen]>0 Then rocGroup  = GroupC / GroupC[relStrLen]

'SCREENING RULES
Dim VolumeRule,TrendRule,VolaRule,RelStrRule,PullBackRule,FundamentalRule
Dim SetUp As BarArray
VolumeRule = avgVolume > minAvgVolume
TrendRule = C > emaLT
If volaAvgINDX>0 Then VolaRule = volaAvg/volaAvgINDX > minVolaRatio
If rocINDX>0 Then RelStrRule = rocStk/rocINDX > rsSPXmin And rocGroup/rocINDX >
rsGroupmin
PullBackRule = rsiWilder < rsiBuyLvl
'FundamentalRule = eps > eps est
SetUp = TrendRule And PullBackRule And VolumeRule And RelStrRule 'And VolaRule
Dim EntryTrigger As BarArray
EntryTrigger = C > Lowest(H,2,1)
If (SetUp[1] Or SetUp) And EntryTrigger Then Buy("LE",1,0,Market,Day)

If (C < Lowest(L,trailBars,1) And C < emaST) Then
    ExitLong("LX_trail","",1,0,Market,Day)
End If
If (C[1] > emaST[1] And C < emaST And countof(C < emaST,BarsSinceEntry,0)>1) Then
    ExitLong("LX_ema","",1,0,Market,Day)
```

```
End If
If rsiWilder > rsiExitLvl Then
    ExitLong("LX_rsi","",1,0,Market,Day)
End If
End Sub
'-------------------------------------------------------------------------------------
-------
'FUNCTION TO COMPUTE VOLATILITY:
Function Volatility(price As BarArray, volaLen) As BarArray
    Dim dp
    Dim dpSqr
    Dim totDpSqr
    Dim sumDp
    Dim sumDpSqAvg
    Dim annualizationFactor

    If price[1] <> 0 Then dp = Log(price/price[1])
    dpSqr = dp*dp
    totDpSqr = Summ(dpSqr, volaLen)
    sumDp = Summ(dp, volaLen)
    if volalen > 0 then sumDpSqAvg = (sumDp*sumDp)/volaLen
    annualizationFactor = 252
    Volatility = 100 * Sqr((annualizationFactor/(volaLen - 1))*(totDpSqr -
sumDpSqAvg))
End Function
'-------------------------------------------------------------------------------------
---------
'HELPER FUNCTION USED BY VOLATILITY FUNCTION TO GET SUMMATION:
' SUMM
' Adds all the values passed to it

Function Summ(Var1 As BarArray,Length) As BarArray
    Dim Counter
    Dim CSum As BarArray

    Counter = 0
    CSum = 0

    CSum = 0
    For Counter = 0 To Length - 1
        CSum = CSum + Var1[Counter]
    Next
    Summ = CSum
End Function
'-------------------------------------------------------------------------------------
-----------
```

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

UPDATA: MAY 2014

This Traders' Tip is based on "A Trading Method For The Long Haul" by Donald W. Pendergast Jr., which appeared in the 2014 Bonus Issue of STOCKS & COMMODITIES.

In the article, the author develops a long-only system for equity products that have beaten their quarterly earnings expectations consistently. The first stage of the process is to filter your universe of stocks using Updata's scan facility to create a list based on that fundamental criteria.

On this list, you apply two exponentially weighted averages, which are used to determine short-term weakness in an overall uptrend. Above average trading volume, outperformance of the SPX index and a two-period RSI are also included as trade filters. The system exits on a close below some trailing-period low. All parameter values can be optimized within the Updata System Optimiser.

The Updata code for this 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 a firewall may paste the code shown below into the Updata custom editor and save it.

```
PARAMETER "RSI Period" #RSIPeriod=2
PARAMETER "EMA Long Period" #EMAPeriodLong=200
PARAMETER "EMA Short Period" #EMAPeriodShort=6
PARAMETER "Volume Avg" #VOLUMEPERIOD=60
PARAMETER "Volume (Millions)" #VOLTHRESH=1
PARAMETER "SPX Ticker" ~SPXIndex=Select
PARAMETER "Beta Period" #BETAPeriod=200
PARAMETER "Beta Thresh." @BETAThresh=1
PARAMETER "Lower High Period" #LOWERHIGHPERIOD=5
NAME "EMA[" #EMAPeriodLong "," #EMAPeriodShort "]" ""
NAME3 "RSI[" #RSIPeriod "]" ""
NAME4 "Beta" ""
NAME6 "Volume" ""
DISPLAYSTYLE 8LINES
INDICATORTYPE TOOL
INDICATORTYPE3 CHART
INDICATORTYPE4 CHART
INDICATORTYPE6 CHART
PLOTSTYLE6 HISTOGRAM
COLOUR RGB(200,0,0)
COLOUR2 RGB(0,0,200)
COLOUR4 RGB(0,0,0)
COLOUR5 RGB(200,0,0)
COLOUR7 RGB(200,0,0)
COLOUR8 RGB(0,0,200)
@RSI=0
@EMALong=0
@EMAShort=0
@BETA=0
@VOLUMEAVG=0
```

```
@STOPLONG=0
@TRAILINGHIGHS=0
@ACCEPTLONGS=0
FOR #CURDATE=MAX(#EMAPeriodLong,#BETAPeriod) TO #LASTDATE-1
   @RSI=RSI(#RSIPeriod)
   @EMALong=EAVE(#EMAPeriodLong)
   @EMAShort=EAVE(#EMAPeriodShort)
   @BETA=BETA(CLOSE,~SPXIndex,#BETAPeriod)
   @VOLUMEAVG=SGNL(VOL,#VOLUMEPERIOD,M)
   'ACCEPTS LONGS AFTER RSI CROSSING BELOW 5
   'LONG SIGNAL ABROGATED WHEN RSI MOVES ABOVE 50
   IF @RSI<5
      @ACCEPTLONGS=1
   ELSEIF @RSI>50
      @ACCEPTLONGS=0
   ENDIF
   'STOP OR EMA EXIT
   IF ORDERISOPEN=1 AND HASX(CLOSE,@EMAShort,DOWN) OR CLOSE<@STOPLONG
      SELL CLOSE
   ENDIF
   IF @TRAILINGHIGHS=1 AND HIGH>HIGH(1) AND ORDERISOPEN=0 AND @ACCEPTLONGS=1 AND
@VOLUMEAVG>(#VOLTHRESH*1000000)
      BUY HIGH(1)
      @STOPLONG=PLOW(LOW(1),3)
   ENDIF
   'LOGS NUMBER OF TRAILING HIGHS
   @TRAILINGHIGHS=SGNL(HIGH<HIGH(1),#LOWERHIGHPERIOD,M)
   'CREATES A TRAILING STOP OF LOWS
   IF ORDERISOPEN=1
      @STOPLONG=MAX(@STOPLONG,PLOW(LOW(1),3))
   ENDIF
   @PLOT=@EMAShort
   @PLOT2=@EMALong
   @PLOT3=@RSI
   @PLOT4=@BETA
   @PLOT5=@BETAThresh
   @PLOT6=VOL
   @PLOT7=#VOLTHRESH*1000000
   @PLOT8=@VOLUMEAVG
NEXT
```

A sample chart is shown in Figure 9.

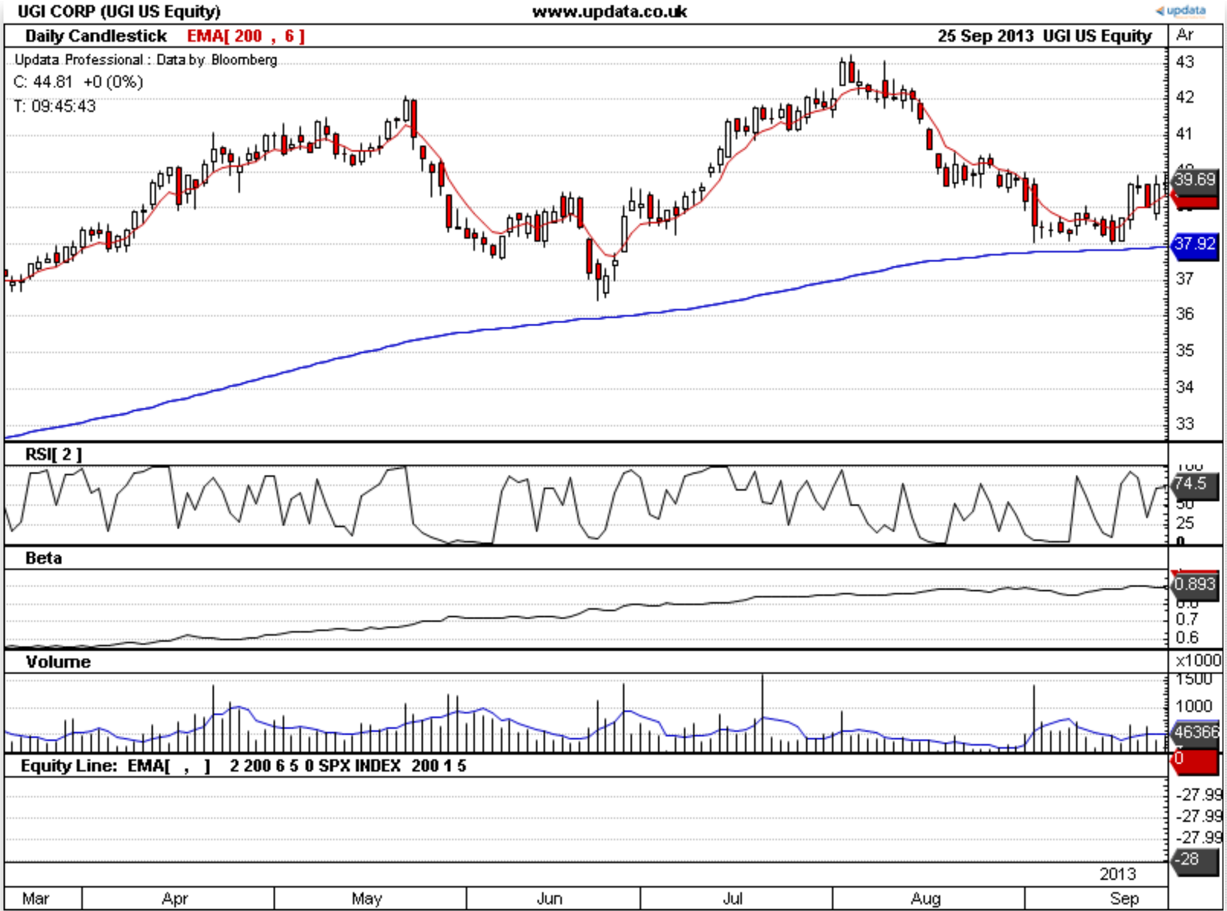


FIGURE 9: UPDATA. This chart shows the trading rules as applied to UGI Corp. data of daily resolution.

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

BACK TO LIST

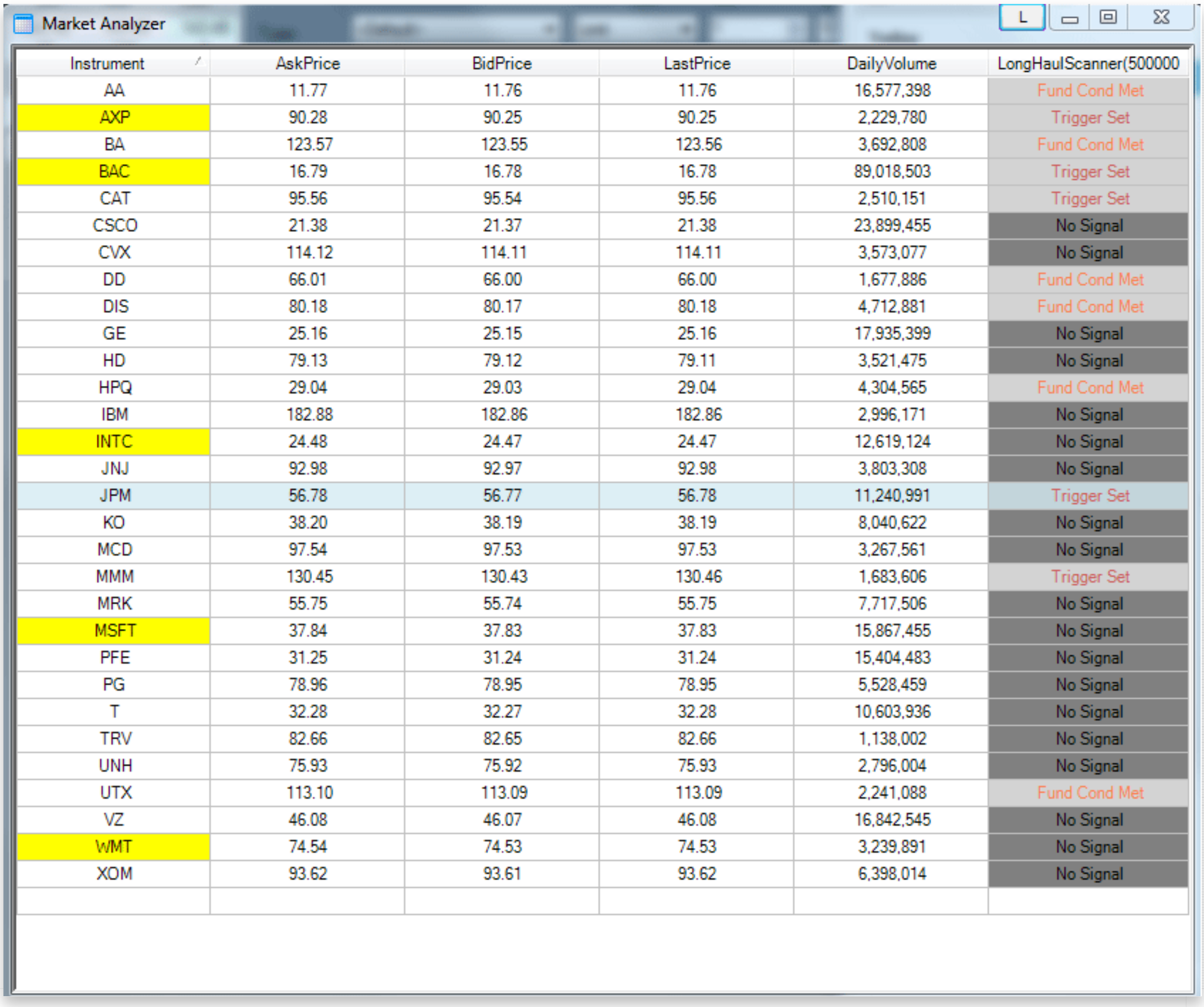
NINJATRADER: MAY 2014

The LongHaulScanner, as discussed in "A Trading Method For The Long Haul" by Donald W. Pendergast Jr. in the 2014 Bonus Issue of STOCKS & COMMODITIES, has been implemented as indicator available for download at www.ninjatrader.com/SC/May2014SC.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 → Strategy from within the NinjaTrader Control Center window and selecting the LongHaulScanner file.

A sample chart implementing the strategy is shown in Figure 10.



Instrument	AskPrice	BidPrice	LastPrice	DailyVolume	LongHaulScanner(500000
AA	11.77	11.76	11.76	16,577,398	Fund Cond Met
AXP	90.28	90.25	90.25	2,229,780	Trigger Set
BA	123.57	123.55	123.56	3,692,808	Fund Cond Met
BAC	16.79	16.78	16.78	89,018,503	Trigger Set
CAT	95.56	95.54	95.56	2,510,151	Trigger Set
CSCO	21.38	21.37	21.38	23,899,455	No Signal
CVX	114.12	114.11	114.11	3,573,077	No Signal
DD	66.01	66.00	66.00	1,677,886	Fund Cond Met
DIS	80.18	80.17	80.18	4,712,881	Fund Cond Met
GE	25.16	25.15	25.16	17,935,399	No Signal
HD	79.13	79.12	79.11	3,521,475	No Signal
HPQ	29.04	29.03	29.04	4,304,565	Fund Cond Met
IBM	182.88	182.86	182.86	2,996,171	No Signal
INTC	24.48	24.47	24.47	12,619,124	No Signal
JNJ	92.98	92.97	92.98	3,803,308	No Signal
JPM	56.78	56.77	56.78	11,240,991	Trigger Set
KO	38.20	38.19	38.19	8,040,622	No Signal
MCD	97.54	97.53	97.53	3,267,561	No Signal
MMM	130.45	130.43	130.46	1,683,606	Trigger Set
MRK	55.75	55.74	55.75	7,717,506	No Signal
MSFT	37.84	37.83	37.83	15,867,455	No Signal
PFE	31.25	31.24	31.24	15,404,483	No Signal
PG	78.96	78.95	78.95	5,528,459	No Signal
T	32.28	32.27	32.28	10,603,936	No Signal
TRV	82.66	82.65	82.66	1,138,002	No Signal
UNH	75.93	75.92	75.93	2,796,004	No Signal
UTX	113.10	113.09	113.09	2,241,088	Fund Cond Met
VZ	46.08	46.07	46.08	16,842,545	No Signal
WMT	74.54	74.53	74.53	3,239,891	No Signal
XOM	93.62	93.61	93.62	6,398,014	No Signal

FIGURE 10: NINJATRADER MARKETANALYZER. This screenshot shows the scanner indicator applied to the DOW30 in NinjaTrader's MarketAnalyzer.

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

BACK TO LIST

TRADERS' TIPS

June 2014

For this month's Traders' Tips, the focus is Perry Kaufman's article in this issue, "Slope Divergence: Capitalizing On Uncertainty." Here we present the June 2014 Traders' Tips code with possible implementations in various software.

Code for TradeStation is already provided in Perry Kaufman's 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. The entries are contributed by various software developers or programmers for software that is capable of customization.

TRADESTATION: JUNE 2014

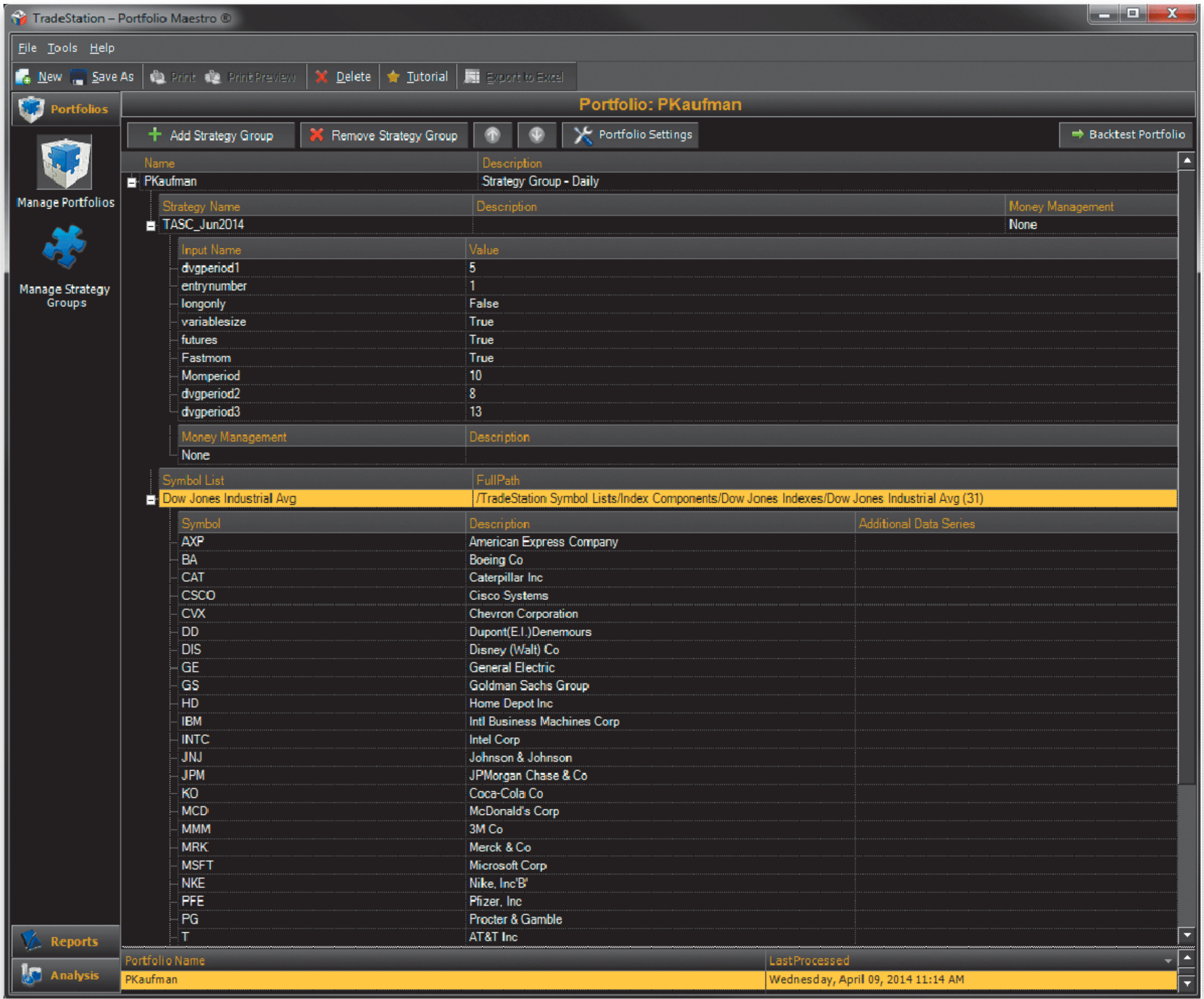
In "Slope Divergence: Capitalizing On Uncertainty" in this issue, author Perry Kaufman presents a trading strategy based on finding a divergence between a security's price and its momentum. The author has already provided some TradeStation Easy-Language code for the strategy in his article in this issue, which we have also made available for download at the link below.

In addition to backtesting the strategy in a Trade-Station chart on one symbol at a time, remember that you can use TradeStation's Portfolio Maestro product to quickly backtest on a portfolio of symbols of your choice. You can also use the TradeStation Walk-Forward Optimizer to automate the complex, multistep task of carrying out a statistical walk-forward test of a trading strategy's optimized inputs.

To download the EasyLanguage code, please visit our TradeStation and EasyLanguage support forum. The code can be found here: http://www.tradestation.com/TASC-2014. The ELD filename is "_TASC_SlopeDivergence.ELD."

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

A sample chart is shown in Figure 1.



Name	Description
PKaufman	Strategy Group - Daily

Strategy Name	Description	Money Management
TASC_Jun2014		None

Input Name	Value
dvgperiod1	5
entrynumber	1
longonly	False
variablesize	True
futures	True
Fastmom	True
Momperiod	10
dvgperiod2	8
dvgperiod3	13

Money Management	Description
None	

Symbol List	FullPath
Dow Jones Industrial Avg	/TradeStation Symbol Lists/Index Components/Dow Jones Indexes/Dow Jones Industrial Avg (31)

Symbol	Description	Additional Data Series
AXP	American Express Company	
BA	Boeing Co	
CAT	Caterpillar Inc	
CSCO	Cisco Systems	
CVX	Chevron Corporation	
DD	Dupont(E.I.)Denemours	
DIS	Disney (Walt) Co	
GE	General Electric	
GS	Goldman Sachs Group	
HD	Home Depot Inc	
IBM	Intl Business Machines Corp	
INTC	Intel Corp	
JNJ	Johnson & Johnson	
JPM	JPMorgan Chase & Co	
KO	Coca-Cola Co	
MCD	McDonald's Corp	
MMM	3M Co	
MRK	Merck & Co	
MSFT	Microsoft Corp	
NKE	Nike, Inc'B'	
PFE	Pfizer, Inc	
PG	Procter & Gamble	
T	AT&T Inc	

Portfolio Name	LastProcessed
PKaufman	Wednesday, April 09, 2014 11:14 AM

FIGURE 1: TRADESTATION. The strategy described here can be backtested and optimized on a portfolio of securities using TradeStation's Portfolio Maestro.

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

MetaStock

METASTOCK: JUNE 2014

In “Slope Divergence: Capitalizing On Uncertainty” in this issue, author Perry Kaufman describes a divergence-based trading system. The formulas for this system are shown here:

```
Buy Order:

tpm:= 25;
tp1:= 5;
tp2:= 12;
tp3:= 14;
mi:= Stoch(tpm, 1);

md1:= linregslope(mi, tp1);
md2:= linregslope(mi, tp2);
md3:= linregslope(mi, tp3);
pd1:= linregslope(c, tp1);
pd2:= linregslope(c, tp2);
pd3:= linregslope(c, tp3);

((md1<0) + (md2<0) + (md3<0) >= 2) AND
((pd1>0) + (pd2>0) + (pd3>0) >= 2)



Sell Order:

tpm:= 25;
tp1:= 5;
tp2:= 12;
tp3:= 14;
mi:= Stoch(tpm, 1);

md1:= linregslope(mi, tp1);
md2:= linregslope(mi, tp2);
md3:= linregslope(mi, tp3);
pd1:= linregslope(c, tp1);
pd2:= linregslope(c, tp2);
pd3:= linregslope(c, tp3);

min(md1, min(md2, min(md3, min(pd1, min(pd2, pd3))))) >0 OR
max(md1, max(md2, max(md3, max(pd1, max(pd2, pd3))))) <0


Sell Short Order:

tpm:= 25;
tp1:= 5;
tp2:= 12;
tp3:= 14;
mi:= Stoch(tpm, 1);
```

```
md1:= linregslope(mi, tp1);
md2:= linregslope(mi, tp2);
md3:= linregslope(mi, tp3);
pd1:= linregslope(c, tp1);
pd2:= linregslope(c, tp2);
pd3:= linregslope(c, tp3);

 ((md1>0) + (md2>0) + (md3>0) >= 2) AND
 ((pd1<0) + (pd2<0) + (pd3<0) >= 2)
```

Buy to CovrOrder:

```
tpm:= 25;
tp1:= 5;
tp2:= 12;
tp3:= 14;
mi:= Stoch(tpm, 1);

md1:= linregslope(mi, tp1);
md2:= linregslope(mi, tp2);
md3:= linregslope(mi, tp3);
pd1:= linregslope(c, tp1);
pd2:= linregslope(c, tp2);
pd3:= linregslope(c, tp3);

min(md1, min(md2, min(md3, min(pd1, min(pd2, pd3))))) >0 OR
max(md1, max(md2, max(md3, max(pd1, max(pd2, pd3))))) <0
```

—William Golson
MetaStock Technical Support
www.metastock.com

BACK TO LIST

THINKORSWIM: JUNE 2014

In “Slope Divergence: Capitalizing On Uncertainty” in this issue, author Perry Kaufman expands on the classic idea of divergence by combining an indicator and price movement. The indicator of focus is stochastics. The article goes into great detail, but the essence is that the divergence signals displayed will normally be missed.

We have made this strategy easy for users to load. Simply click on the link http://tos.mx/WP7mBx and choose *backtest in thinkorswim*. You can adjust the parameters of these within the *edit studies* window to fine-tune your variables.

The chart in Figure 2 shows the entry & exit point when the criteria described in the article were met on a one-year daily chart of Microsoft (MSFT). What is being investigated here and what is triggering the backtested trades are the slopes of the stochastics and the slopes of the price movement. This strategy calculates multiple lengths and bases the entry points on the lengths that meet Kaufman's criteria.

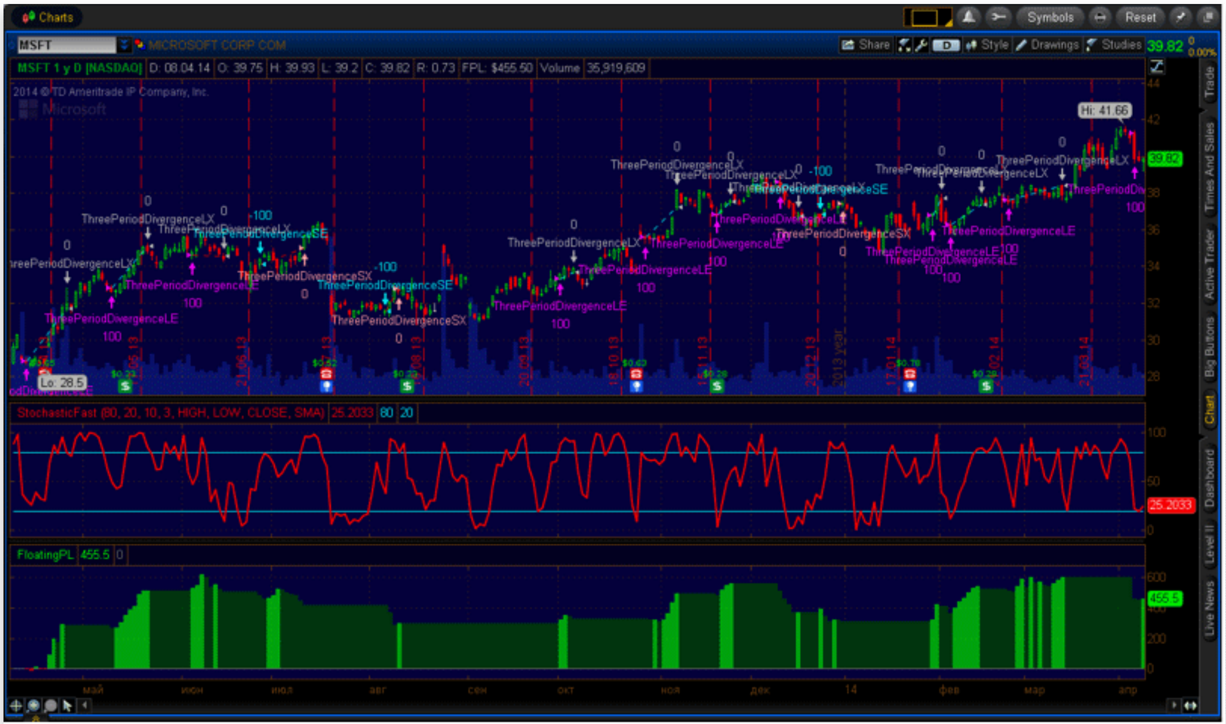


FIGURE 2: THINKORSWIM

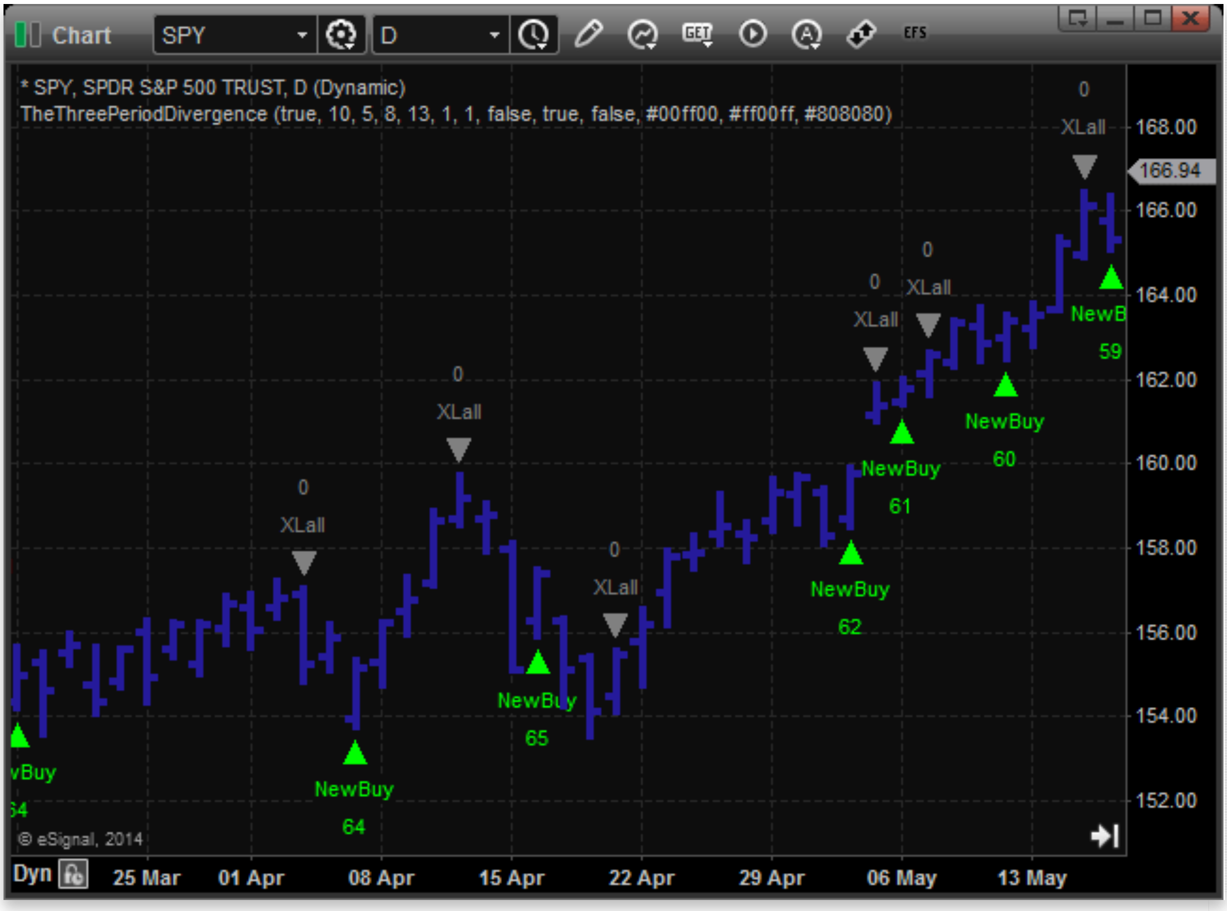
For more information about the strategy, refer to Kaufman's article in this issue. Happy swimming!

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

BACK TO LIST

eSIGNAL: JUNE 2014

For this month's Traders' Tip, we've provided the formula TheThreePeriodDivergence.efs based on the formula described in Perry Kaufman's article in this issue, "Slope Divergence: Capitalizing On Uncertainty." 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 implementation of the technique is shown in Figure 3.



* SPY, SPDR S&P 500 TRUST, D (Dynamic)
TheThreePeriodDivergence (true, 10, 5, 8, 13, 1, 1, false, true, false, #00ff00, #ff00ff, #808080)

FIGURE 3: eSIGNAL

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 download here.

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

BACK TO LIST

WealthLab.com

WEALTH-LAB: JUNE 2014

This WealthScript strategy implements a simplified version of Perry Kaufman's triple-divergence trading system described in his article in this issue, "Slope Divergence: Capitalizing On Uncertainty." Users have the control to turn the system into long-only, as well as to configure its sensitivity by triggering a trading signal after one, two, or three divergences on the same bar.

C# Code

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

namespace WealthLab.Strategies
{
        enum Side { Buy = 1, Sell = -1, NA = 0 }

        public class TASC201406_Kaufman : WealthScript
        {
                Side SingleDivergence( int bar, int period, int momperiod,
                        DataSeries ps, DataSeries ms, out bool trendup, out bool
trenddn, out int p, out int m )
                {
                        trendup = (ps[bar] > 0 && ms[bar] > 0);
                        trenddn = (ps[bar] < 0 && ms[bar] < 0);
                        p = ps[bar] >= 0 ? 1 : ps[bar] < 0 ? -1 : 0;
                        m = ms[bar] >= 0 ? 1 : ms[bar] < 0 ? -1 : 0;

                        return ( ps[bar] > 0 && ms[bar] < 0 ) ? Side.Buy :
                                ( ps[bar] < 0 && ms[bar] > 0 ) ? Side.Sell : Side.NA;
                }

                void Colorize( int bar, Side side )
                {
                        SetBackgroundColor( bar, Color.FromArgb(30, side == Side.Buy
? Color.Green : Color.Red ) );
                }

                private StrategyParameter paramNumDiv;
                private StrategyParameter paramLongOnly;

                public TASC201406_Kaufman()
                {
                        paramNumDiv = CreateParameter("Divergences", 1, 1, 3, 1);
                        paramLongOnly = CreateParameter("Long Only", 0, 0, 1, 1);
                }

                protected override void Execute()
                {
                        int momperiod = 10, entrynumber = paramNumDiv.ValueInt,
                                dvgperiod1 = 5, dvgperiod2 = 8, dvgperiod3 = 13,
maxdivergences = 3;
                        bool longonly = (paramLongOnly.ValueInt == 0 ? false : true);
                        LinearRegSlope lsp1 = LinearRegSlope.Series( Close,dvgperiod1
);
                        LinearRegSlope lsp2 = LinearRegSlope.Series( Close,dvgperiod2
);
                        LinearRegSlope lsp3 = LinearRegSlope.Series( Close,dvgperiod3
);
                        StochK fastK = StochK.Series(Bars,momperiod);
                        LinearRegSlope lsm1 = LinearRegSlope.Series( fastK,dvgperiod1
);
                        LinearRegSlope lsm2 = LinearRegSlope.Series( fastK,dvgperiod2
);
```

```
            LinearRegSlope lsm3 = LinearRegSlope.Series( fastK,dvgperiod3
);

            ClearDebug(); HideVolume(); LineStyle ls = LineStyle.Solid;
            ChartPane lrspPane = CreatePane( 40,false,true );
            PlotSeries( lrspPane, lsp1, Color.DarkGreen, ls, 1 );
            PlotSeries( lrspPane, lsp2, Color.Blue, ls, 1 );
            PlotSeries( lrspPane, lsp3, Color.Red, ls, 1 );
            //ChartPane lrsmPane = CreatePane( 30,false,true );
            //PlotSeries( lrsmPane, fastK, Color.Black, ls, 1 );

            for(int bar =
GetTradingLoopStartBar(Math.Max(momperiod,dvgperiod3)); bar < Bars.Count; bar++)
            {
                int nbuys = 0, nsells = 0, ps1 = 0, ps2 = 0, ps3 = 0,
ms1 = 0, ms2 = 0, ms3 = 0,
                    npriceslopeup = 0, npriceslopedown = 0,
nmomslopeup = 0, nmomslopedown = 0;
                bool trend1up = false, trend2up = false, trend3up =
false,
                    trend1dn = false, trend2dn = false, trend3dn
= false;

                Side _d1 = SingleDivergence( bar, dvgperiod1,
momperiod, lsp1, lsm1, out trend1up, out trend1dn, out ps1, out ms1 );
                Side _d2 = SingleDivergence( bar, dvgperiod2,
momperiod, lsp2, lsm2, out trend2up, out trend2dn, out ps2, out ms2 );
                Side _d3 = SingleDivergence( bar, dvgperiod3,
momperiod, lsp3, lsm3, out trend3up, out trend3dn, out ps3, out ms3 );

                int d1 = (int)_d1;
                int d2 = (int)_d2;
                int d3 = (int)_d3;

                if( d1 > 0 ) nbuys += 1; if( d1 < 0 ) nsells += 1;
                if( d2 > 0 ) nbuys += 1; if( d2 < 0 ) nsells += 1;
                if( d3 > 0 ) nbuys += 1; if( d3 < 0 ) nsells += 1;

                if( ps1 > 0 ) npriceslopeup += 1; if( ps1 < 0 )
npriceslopedown += 1;
                if( ps2 > 0 ) npriceslopeup += 1; if( ps2 < 0 )
npriceslopedown += 1;
                if( ps3 > 0 ) npriceslopeup += 1; if( ps3 < 0 )
npriceslopedown += 1;

                if( ms1 > 0 ) nmomslopeup += 1; if( ms1 < 0 )
nmomslopedown += 1;
                if( ms2 > 0 ) nmomslopeup += 1; if( ms2 < 0 )
nmomslopedown += 1;
                if( ms3 > 0 ) nmomslopeup += 1; if( ms3 < 0 )
nmomslopedown += 1;

                if( nbuys >= entrynumber )
                    Colorize( bar, Side.Buy );
                else
                    if( nsells >= entrynumber )
                        Colorize( bar, Side.Sell );
                if( false )
                PrintDebug( "bar: " + bar + ", d1: " + d1 + ", d2: "
+ d2 + ", d3: " + d3 + ", nbuys: " + nbuys + ", nsells: " + nsells +
                                        ", npriceslopeup: " + npriceslopeup + ",
npriceslopedown: " + npriceslopedown +
                                        ", nmomslopeup: " + nmomslopeup + ",
nmomslopedown: " + nmomslopedown );

                                if (IsLastPositionActive)
                                {
                                        Position p = LastPosition;

                                        if( p.PositionType == PositionType.Long )
                                        {
                                                if( nsells >= entrynumber)
                                                        SellAtClose( bar, p,
"revLong");
                                        }
                                        else //if( p.PositionType ==
PositionType.Short )
                                        {
                                                if( nbuys >= entrynumber)
                                                        CoverAtClose( bar, p,
"revShort");
                                        }

                                        if( (npriceslopeup == maxdivergences &&
nmomslopeup == maxdivergences) ||
                                                (npriceslopedown == maxdivergences &&
nmomslopedown == maxdivergences) )
                                        {
                                                ExitAtMarket( bar+1, p,
p.PositionType == PositionType.Long ? "xLall" : "xSall" );
                                        }
                                }
                                else
                                {
                                        if( nbuys >= entrynumber)
                                                BuyAtMarket(bar+1, "newBuy");
                                        else
                                                if( nsells >= entrynumber && longonly
== false)
                                                ShortAtMarket(bar+1, "newSell");
                                }
                        }
                }
        }
}
```

A sample chart implementation is shown in Figure 4.

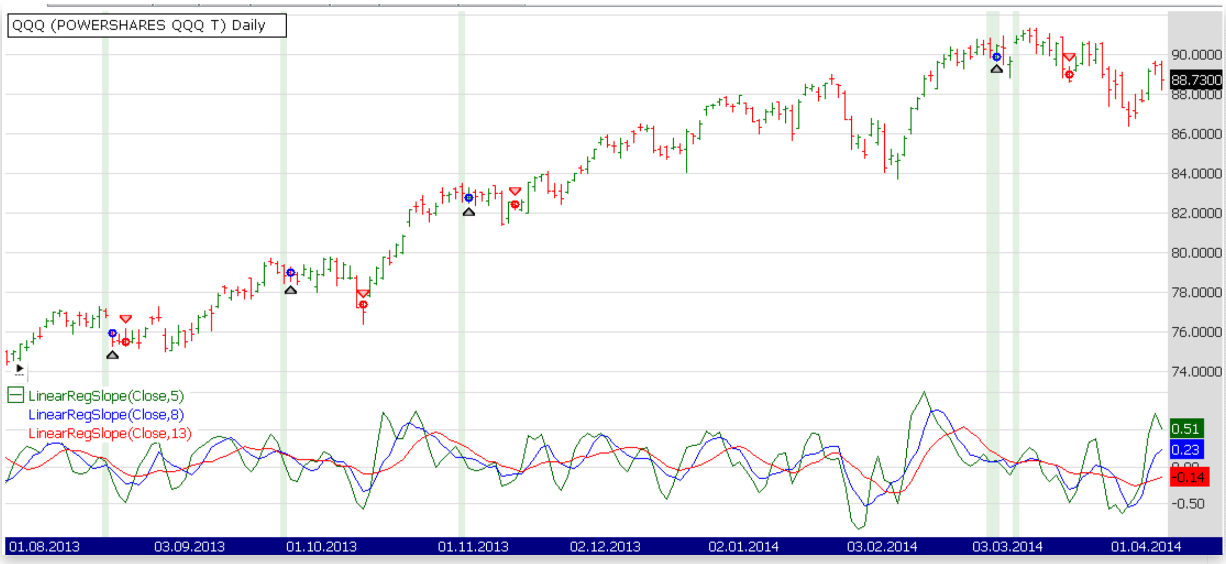


FIGURE 4: WEALTH-LAB, QQQ. This chart illustrates the application of the system's rules on a daily chart of QQQ. The lower panel shows the five-, eight-, and 13-day linear regression slope of the recent close price.

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

BACK TO LIST

AMIBROKER: JUNE 2014

In "Slope Divergence: Capitalizing On Uncertainty" in this issue, author Perry Kaufman presents a strategy based on divergences between stochastics and price action. A ready-to-use formula based on the article for use in AmiBroker is presented here. To use the formula, copy it to the AFL formula editor, then press the send to analysis button. From the analysis window, you can press backtest to evaluate trading system performance.

A sample chart implemention is shown in Figure 5.

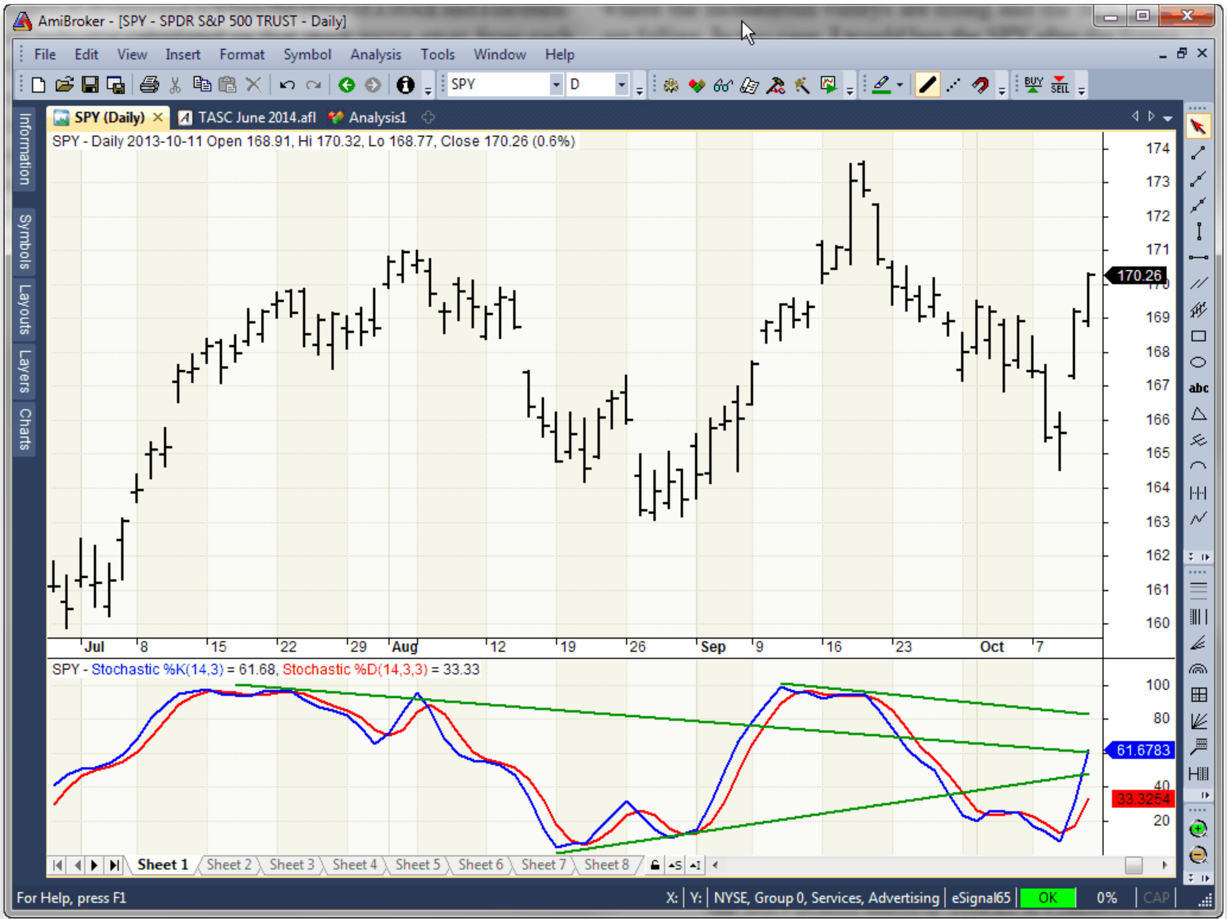

AmiBroker - [SPY - SPDR S&P 500 TRUST - Daily]
File Edit View Insert Format Symbol Analysis Tools Window Help
SPY
D
BUY
SELL
Information
Symbols
Layouts
Layers
Charts
SPY (Daily)
TASC June 2014.afl
Analysis1
SPY - Daily 2013-10-11 Open 168.91, Hi 170.32, Lo 168.77, Close 170.26 (0.6%)
174
173
172
171
170.26
169
168
167
166
165
164
163
162
161
160
Jul
8
15
22
29
Aug
12
19
26
Sep
9
16
23
Oct
7
SPY - Stochastic %K(14,3) = 61.68, Stochastic %D(14,3,3) = 33.33
100
80
61.6783
40
33.3254
20
Sheet 1
Sheet 2
Sheet 3
Sheet 4
Sheet 5
Sheet 6
Sheet 7
Sheet 8
For Help, press F1
X:
Y:
NYSE, Group 0, Services, Advertising
eSignal65
OK
0%
CAP

FIGURE 5: AMIBROKER. Here is a daily chart of the SPY with a 14-day slow stochastic showing divergence with price action.

LISTING 1.

```
fastmom = ParamToggle("Fast momenum", "No|Yes", 1 );
momperiod = Param("Momentum period", 10, 1, 100 );;
dvgperiod1 = Param("Div. period 1", 5, 1, 100 );
dvgperiod2 = Param("Div. period 2", 8, 1, 100 );;
dvgperiod3 = Param("Div. period 3", 13, 1, 100 );
ATRperiod = Param("ATR period", 20, 1, 1, 100 );
entrynumber = Param("Entry no", 1, 1, 3, 1 );
maxdivergences = 3;
investment = Param("Investment per trade", 25000, 10000, 250000, 1000 );

mom = IIf( fastmom, StochK( momperiod, 1 ), StochK( momperiod, 3 ) );

priceslope1 = LinRegSlope(Close,dvgperiod1);
momslope1 = LinRegSlope(mom,dvgperiod1);
dvgbuy1 = priceslope1 > 0 AND momslope1 < 0;
dvgsell1 = priceslope1 < 0 AND momslope1 > 0;

priceslope2 = LinRegSlope(Close,dvgperiod2);
momslope2 = LinRegSlope(mom,dvgperiod2);
dvgbuy2 = priceslope2 > 0 AND momslope2 < 0;
dvgsell2 = priceslope2 < 0 AND momslope2 > 0;

priceslope3 = LinRegSlope(Close,dvgperiod3);
```

```
momslope3 = LinRegSlope(mom,dvgperiod3);
dvgbuy3 =  priceslope3 > 0 AND momslope3 < 0;
dvgsell3 = priceslope3 < 0 AND momslope3 > 0;

nbuys = dvgbuy1 + dvgbuy2 + dvgbuy3;
nsells = dvgsell1 + dvgsell2 + dvgsell3;

npriceslopeup = ( priceslope1 >= 0 ) + ( priceslope2 >= 0 ) + ( priceslope3 >= 0 );
npriceslopedown = ( priceslope1 < 0 ) + ( priceslope2 < 0 ) + ( priceslope3 < 0 );

nmomslopeup = ( momslope1 >= 0 ) + ( momslope2 >= 0 ) + ( momslope3 >= 0 );
nmomslopedown = ( momslope1 < 0 ) + ( momslope2 < 0 ) + ( momslope3 < 0 );

Exit = ( npriceslopeup == maxdivergences AND nmomslopeup == maxdivergences) OR (
npriceslopedown == maxdivergences AND nmomslopedown == maxdivergences);
Cover = Sell = Exit; // exit both long and short

Buy = nbuys >= entrynumber;
Short = nsells >= entrynumber;

SetOption("ReverseSignalForcesExit", True ); // to reverse position

SetPositionSize( investment, spsValue );
```

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

BACK TO LIST

NEUROSHELL TRADER: JUNE 2014

The trading system based on slope divergence presented by Perry Kaufman in his article in this issue, “Slope Divergence: Capitalizing On Uncertainty,” can be easily implemented in NeuroShell Trader using 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 (two of which must be true):

And2( A>B( LinRegSlope(Close,5), 0), A<B( LinRegSlope( Stoch%K(High,Low,Close,10),
5), 0))
And2( A>B( LinRegSlope(Close,8), 0), A<B( LinRegSlope( Stoch%K(High,Low,Close,10),
8), 0))
And2( A>B( LinRegSlope(Close,13), 0), A<B( LinRegSlope( Stoch%K(High,Low,Close,10),
13), 0))

SELL-SHORT CONDITIONS (two of which must be true):

And2( A<B( LinRegSlope(Close,5), 0), A>B(LinRegSlope( Stoch%K(High,Low,Close,10), 5),
0))
And2( A<B( LinRegSlope(Close,8), 0), A>B(LinRegSlope( Stoch%K(High,Low,Close,10), 8),
0))
And2( A<B( LinRegSlope(Close,13), 0), A>B(LinRegSlope( Stoch%K(High,Low,Close,10),
13), 0))
```

SELL LONG CONDITIONS and COVER SHORT CONDITIONS (one of which must be true):

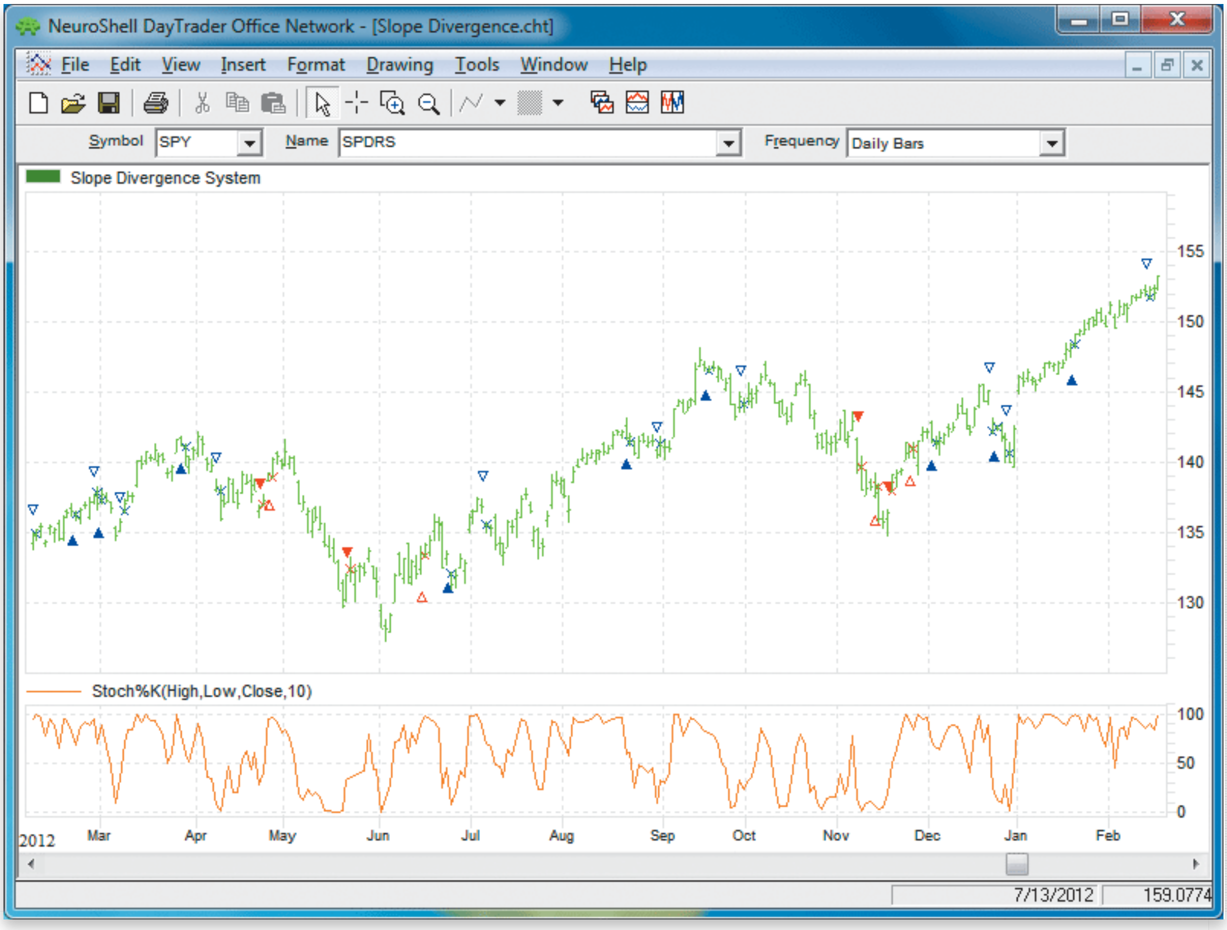
```
And3( And2( A>B( LinRegSlope(Close,5), 0), A>B( LinRegSlope(
Stoch%K(High,Low,Close,10), 5), 0)), And2( A>B( LinRegSlope(Close,8), 0), A>B(
LinRegSlope( Stoch%K(High,Low,Close,10), 8), 0)),
And2( A>B( LinRegSlope(Close,13), 0), A>B( LinRegSlope( Stoch%K(High,Low,Close,10),
13), 0)))

And3( And2( A<B( LinRegSlope(Close,5), 0), A<B( LinRegSlope(
Stoch%K(High,Low,Close,10), 5), 0)), And2( A<B( LinRegSlope(Close,8), 0), A<B(
LinRegSlope( Stoch%K(High,Low,Close,10), 8), 0)),
And2( A<B( LinRegSlope(Close,13), 0), A<B( LinRegSlope( Stoch%K(High,Low,Close,10),
13), 0)))
```

Select also “Long/Short entries exit existing short/long positions” in the trading strategy parameters dialog.

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 website to download a copy of this or any previous Traders’ Tip.

A sample chart is shown in Figure 6.



Symbol SPY
Name SPDRS
Frequency Daily Bars

FIGURE 6: NEUROSHELL TRADER. This NeuroShell Trader chart displays the slope divergence trading system.

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

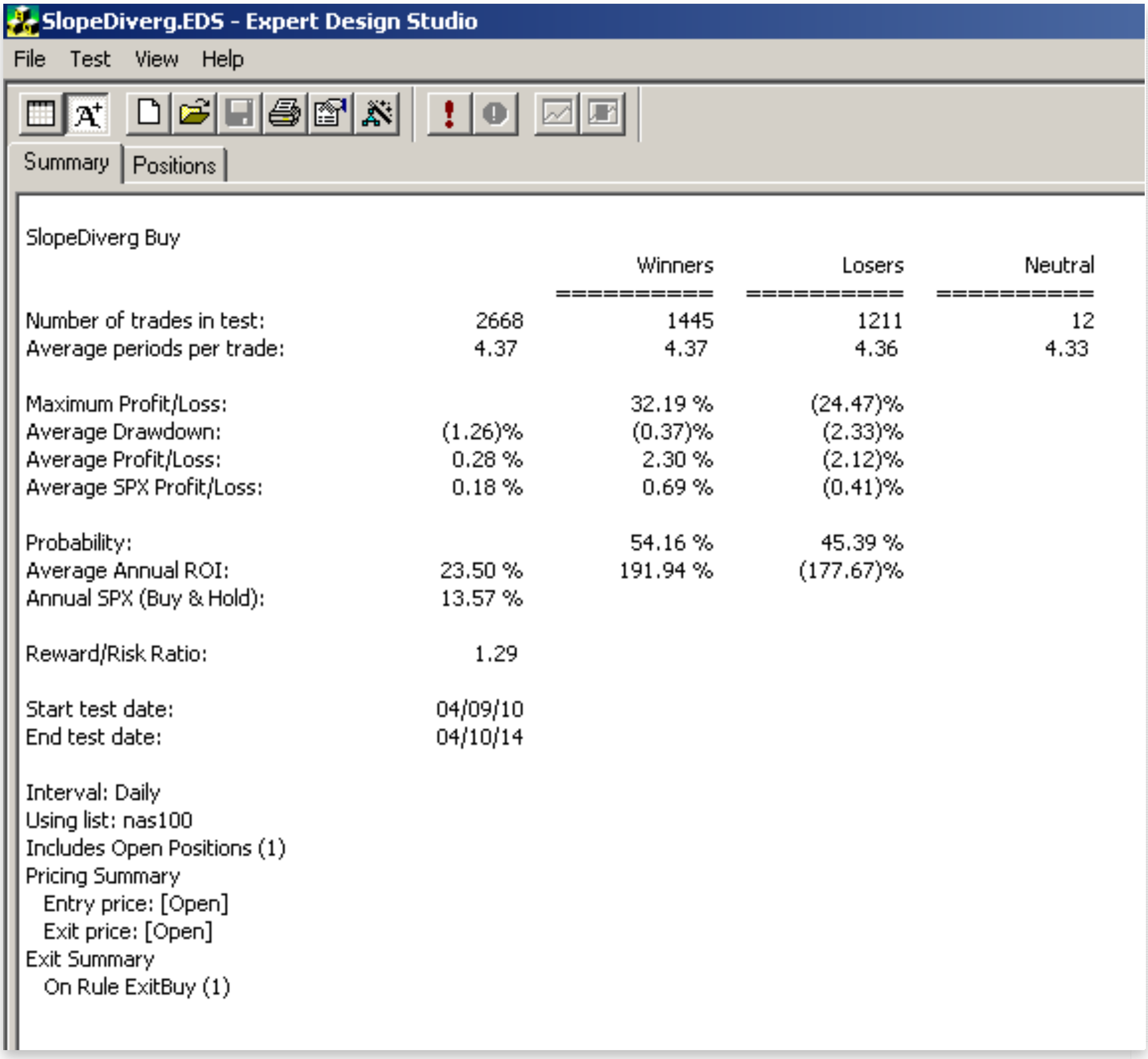
BACK TO LIST

AIQ: JUNE 2014

The AIQ code based on Perry Kaufman's article in this issue, "Slope Divergence: Capitalizing On Uncertainty," is provided at www.TradersEdgeSystems.com/traderstips.htm.

I have modified the implementation somewhat from the author's descriptions. I did not find that the system was exiting in an average of six days but was holding for a longer period. My exits might be the issue so I added a time exit that can be used to force an exit after the “maxBars” input number of bars. I liked the results when my time exit was set to hold for a maximum of nine bars.

Figure 7 shows the AIQ EDS summary long-only backtest report using the NASDAQ 100 list of stocks over the prior four years ending 4/10/2014. Neither commission nor slippage have been subtracted from these results. To get the short side of the system to show a profit, I added slope filters on the NASDAQ 100 index. Note that my parameter settings differ from those suggested by the author.



```
SlopeDiverg Buy
                                           Winners        Losers       Neutral
                                        ==========    ==========    ==========
Number of trades in test:         2668        1445          1211            12
Average periods per trade:        4.37        4.37          4.36          4.33

Maximum Profit/Loss:                       32.19 %      (24.47)%
Average Drawdown:              (1.26)%     (0.37)%       (2.33)%
Average Profit/Loss:            0.28 %      2.30 %       (2.12)%
Average SPX Profit/Loss:        0.18 %      0.69 %       (0.41)%

Probability:                               54.16 %       45.39 %
Average Annual ROI:            23.50 %    191.94 %     (177.67)%
Annual SPX (Buy & Hold):       13.57 %

Reward/Risk Ratio:                1.29

Start test date:              04/09/10
End test date:                04/10/14

Interval: Daily
Using list: nas100
Includes Open Positions (1)
Pricing Summary
  Entry price: [Open]
  Exit price: [Open]
Exit Summary
  On Rule ExitBuy (1)
```

FIGURE 7: AIQ, SAMPLE RESULTS. Here is a sample AIQ EDS summary long-only backtest report using the NASDAQ 100 list of stocks over the prior four years ending 4/10/2014.

The code and EDS file can be downloaded from www.TradersEdgeSystems.com/traderstips.htm. The code is also shown here:

```
!SLOPE DIVERGENCE: CAPITALIZING ON UNCERTAINTY
!Author: Perry Kaufman, TASC June 2014
!Coded by: Richard Denning 4/7/2014
!www.TradersEdgeSystems.com

!INPUTS:
momLen is 10.
dvgLen1 is 5.
dvgLen2 is 7.
dvgLen3 is 10.
entryNum is 3.
maxDiverg is 3.
minPrice is 10.
maxBars is 3.

!USER DEFINED FORMULAS:
C is [close].
L is [low].
H is [high].
HH is highresult(H,momLen).
LL is lowresult(L,momLen).
stoch is (C - LL) / (HH - LL).

momSlope1 is slope2(stoch,dvgLen1).
momSlope2 is slope2(stoch,dvgLen2).
momSlope3 is slope2(stoch,dvgLen3).

priceSlope1 is slope2(C,dvgLen1).
priceSlope2 is slope2(C,dvgLen2).
priceSlope3 is slope2(C,dvgLen3).

dvgBuy1 if priceSlope1 > 0 and momslope1 < 0.
dvgBuy2 if priceSlope1 > 0 and momslope2 < 0.
dvgBuy3 if priceSlope1 > 0 and momslope3 < 0.

dvgSell1 if priceSlope1 < 0 and momslope1 > 0.
dvgSell2 if priceSlope1 < 0 and momslope2 > 0.
dvgSell3 if priceSlope1 < 0 and momslope3 > 0.

nPriceSUp is priceSlope1 > 0 + priceSlope2 > 0 + priceSlope3 > 0.
nMomSUp is momSlope1 > 0 + momSlope2 > 0 + momSlope3 > 0.

nPriceSDown is priceSlope1 < 0 + priceSlope2 < 0 + priceSlope3 < 0.
nMomSDown is momSlope1 < 0 + momSlope2 < 0 + momSlope3 < 0.

dvgBuySum is dvgBuy1 + dvgBuy2 + dvgBuy3.
dvgSellSum is dvgSell1 + dvgSell2 +dvgSell3.

Buy if dvgBuySum >= entryNum and C > minPrice.
AllComboExit if (nPriceSDown = maxDiverg and nMomSDown = maxDiverg)
             or (nPriceSUp = maxDiverg and nMomSUp = maxDiverg).
Time if {position days} >= maxBars.
ExitBuy if AllComboExit or Sell or Time.

Sell if dvgSellSum >= entryNum.
ExitSell if AllComboExit or Buy or Time.
```

—Richard Denning
info@TradersEdgeSystems.com
for AIQ Systems

BACK TO LIST

TRADERSSTUDIO: JUNE 2014

The TradersStudio code based on Perry Kaufman’s article in this issue, “Slope Divergence: Capitalizing On Uncertainty,” 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:

- System SDIVERG_SYS: A trading system that employs most of the rules outlined in Kaufman’s article;
- Function LinearRegSRV: Returns the linear regression (LR) slope and other related LR variables.

The code I am providing departs somewhat from Kaufman’s descriptions. I did not find that the system was exiting in an average of six days but was holding for a longer period. My exits could be the issue, so I added a time exit that can be used to force an exit after the “maxBars” input number of bars. I liked the results when my time exit was set to hold for a maximum of nine bars.

In Figure 8, I show the equity and underwater equity curves based on this system trading the NASDAQ 100 list of stocks, long only, for the period 2008 through February 2014. On each signal, 100 shares were traded. Neither commission nor slippage was subtracted from these results. I did not code or test the short side of the system.

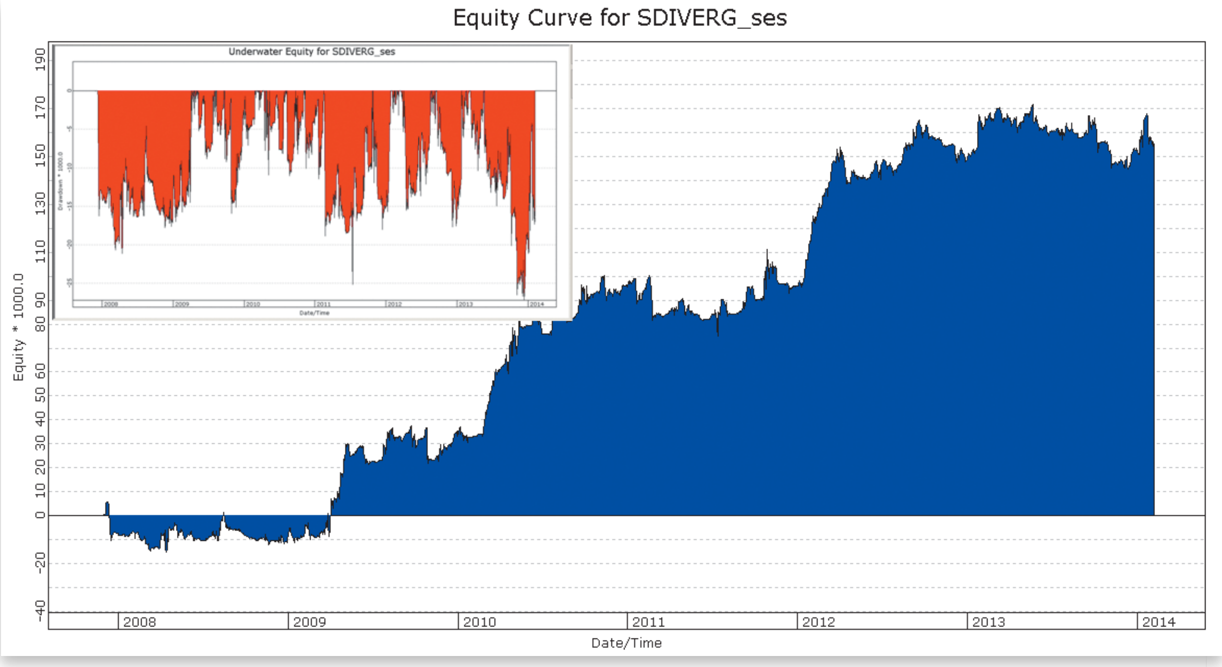


FIGURE 8: TRADERSSTUDIO. Here are sample equity and underwater equity curves trading the NASDAQ 100 list of stocks, long only, for the period 2008 through February 2014.

The code is also shown below:

```
'SLOPE DIVERGENCE: CAPITALIZING ON UNCERTAINTY
'Author: Perry Kaufman, TASC June 2014
'Coded by: Richard Denning 4/7/2014
'www.TradersEdgeSystems.com

Sub SDIVERG_SYS(momLen,dvgLen1,dvgLen2,dvgLen3,entryNum,maxDiverg,minPrice,maxBars)
'momLen=10,dvgLen1=5,dvgLen2=7,dvgLen3=10,entryNum=3,maxDiverg=3,minPrice=10,maxBars=3
)
Dim stoc As BarArray
Dim momSlope1 As BarArray
Dim momSlope2 As BarArray
Dim momSlope3 As BarArray
Dim priceSlope1 As BarArray
Dim priceSlope2 As BarArray
Dim priceSlope3 As BarArray
Dim rSqrd1,rSqrd2,rSqrd3,rSqrd4,rSqrd5,rSqrd6
Dim slope1,slope2,slope3,slope4,slope5,slope6
Dim endVal1,endVal2,endVal3,endVal4,endVal5,endVal6
Dim dvgBuy1,dvgBuy2,dvgBuy3,dvgSell1,dvgSell2,dvgSell3
Dim nPriceSUp,nMomSUp,nPriceSDown,nMomSDown,dvgBuySum,dvgSellSum

If (Highest(H,momLen) - Lowest(L,momLen)) <> 0 Then
   stoc = (C-Lowest(L,momLen))/(Highest(H,momLen) - Lowest(L,momLen))
End If

momSlope1 = LinearRegSRV(stoc,dvgLen1,0,rSqrd1,slope1,endVal1)
momSlope2 = LinearRegSRV(stoc,dvgLen2,0,rSqrd2,slope2,endVal2)
momSlope3 = LinearRegSRV(stoc,dvgLen3,0,rSqrd3,slope3,endVal3)
```

```
priceSlope1 = LinearRegSRV(C,dvgLen1,0,rSqrd4,slope4,endVal4)
priceSlope2 = LinearRegSRV(C,dvgLen2,0,rSqrd5,slope5,endVal5)
priceSlope3 = LinearRegSRV(C,dvgLen3,0,rSqrd6,slope6,endVal6)

dvgBuy1 = IIF(priceSlope1 > 0 And momSlope1 < 0,1,0)
dvgBuy2 = IIF (priceSlope1 > 0 And momSlope2 < 0,1,0)
dvgBuy3 = IIF (priceSlope1 > 0 And momSlope3 < 0,1,0)

dvgSell1 = IIF (priceSlope1 < 0 And momSlope1 > 0,1,0)
dvgSell2 = IIF (priceSlope1 < 0 And momSlope2 > 0,1,0)
dvgSell3 = IIF (priceSlope1 < 0 And momSlope3 > 0,1,0)

nPriceSUp = IIF(priceSlope1 > 0,1,0) + IIF(priceSlope2 > 0,1,0) + IIF(priceSlope3 >
0,1,0)
nMomSUp = IIF(momSlope1 > 0,1,0) + IIF(momSlope2 > 0,1,0) + IIF(momSlope3 > 0,1,0)

nPriceSDown = IIF(priceSlope1 < 0,1,0) + IIF(priceSlope2 < 0,1,0) + IIF(priceSlope3 <
0,1,0)
nMomSDown = IIF(momSlope1 < 0,1,0) + IIF(momSlope2 < 0,1,0) +  IIF(momSlope3 < 0,1,0)

dvgBuySum = dvgBuy1 + dvgBuy2 + dvgBuy3
dvgSellSum = dvgSell1 + dvgSell2 +dvgSell3

If (dvgBuySum >= entryNum And TSCLose > minPrice) Then Buy("LE",1,0,Market,Day)
If (nPriceSDown = maxDiverg And nMomSDown = maxDiverg) Then
ExitLong("LX_allDown","",1,0,Market,Day)
If (nPriceSUp = maxDiverg And nMomSUp = maxDiverg) Then
ExitLong("LX_allUp","",1,0,Market,Day)
If (BarsSinceEntry > maxBars) Then ExitLong("LX_time","",1,0,Market,Day)
If (dvgSellSum >= entryNum) Then ExitLong("LX_sell","",1,0,Market,Day)

End Sub
'---------------------------------------------------------------------------------------
---------------
' LINEAR REGRESSION FUNCTION
' Coded by: Richard Denning 01/21/08

'Parameters

' Y  specifies which Price Of the asset Of interest is To be used
' SLen the number Of trailing bars To consider
' TargetB represents the number Of bars into the future Or back into the past
' Returns a numeric value containing the current value Of the specified regression
line at TargetB.
' Changes values of variables rSqrd, slopeR, endVal to those the least squares line
computed by the function
' R squared (rSqrd) is the measure of how well the line fits the data (will vary from
0 (no fit) to 1.00 (perfect fit)
' slope (slopeR) is the rise over run of the line
' endVal is the value of the line at the current bar

'the regression formulas can be checked using the Excel tutorial on linear regression
found at:
'http://phoenix.phys.clemson.edu/tutorials/excel/regression.html

Function LinearRegSRV(Y As BarArray, SLen, TargetB, ByRef rSqrd, ByRef slopeR, ByRef
endVal) As BarArray
    Dim X As BarArray
    Dim Num1 As BarArray
    Dim Num2 As BarArray
    Dim SumX As BarArray
    Dim SumSqrX As BarArray
    Dim SumY As BarArray
    Dim SumSqrY As BarArray
    Dim SumXY As BarArray
    Dim Slope As BarArray
    Dim Intercept As BarArray

 If SLen <= 0 Then
    LinearRegSRV = 0
 Else

    SumX = 0
    SumSqrX = 0
    SumY = 0
    SumSqrY = 0
    SumXY = 0
    'Sum2 = 0

    For X = 0 To SLen - 1
        SumX = SumX + X
        SumSqrX = SumSqrX + X * X
        SumY = SumY + Y[X]
        SumSqrY = SumSqrY + Y[X] * Y[X]
        SumXY = SumXY + X * Y[X]
    Next

  'slope
    If (SumSqrX - SumSqrY) <> 0 Then
        'Slope = Num1 / Num2
        Slope = (SumXY - SumX * SumY) / (SumSqrX - SumSqrY)
    Else
        Slope = 0
    End If

  'intercept
    Intercept = (SumY - Slope * SumX) / SLen
    slopeR = Slope

  'R squared
    If (Sqr((SumSqrX - SumX*SumX)*(SumSqrY - SumY * SumY))) <> 0 Then
        rSqrd = (SumXY - SumX * SumY) / (Sqr((SumSqrX - SumX*SumX)*(SumSqrY - SumY *
SumY)))
    Else
        rSqrd = 0
    End If

  'end value of linear regression line
    endVal = Intercept + Slope * (SLen - 1)

  'projected value of linear regression line at target bar
    LinearRegSRV = Intercept + Slope * (SLen - 1 - TargetB)

 End If
End Function
```

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

NINJATRADER: JUNE 2014

The slope divergence strategy presented by Perry Kaufman in his article in this issue, “Slope Divergence: Capitalizing On Uncertainty,” has been implemented as a NinjaTrader strategy available for download at www.ninjatrader.com/SC/June2014SC.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 “SlopeDivergence” file.

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

A sample chart implementing the strategy is shown in Figure 9.

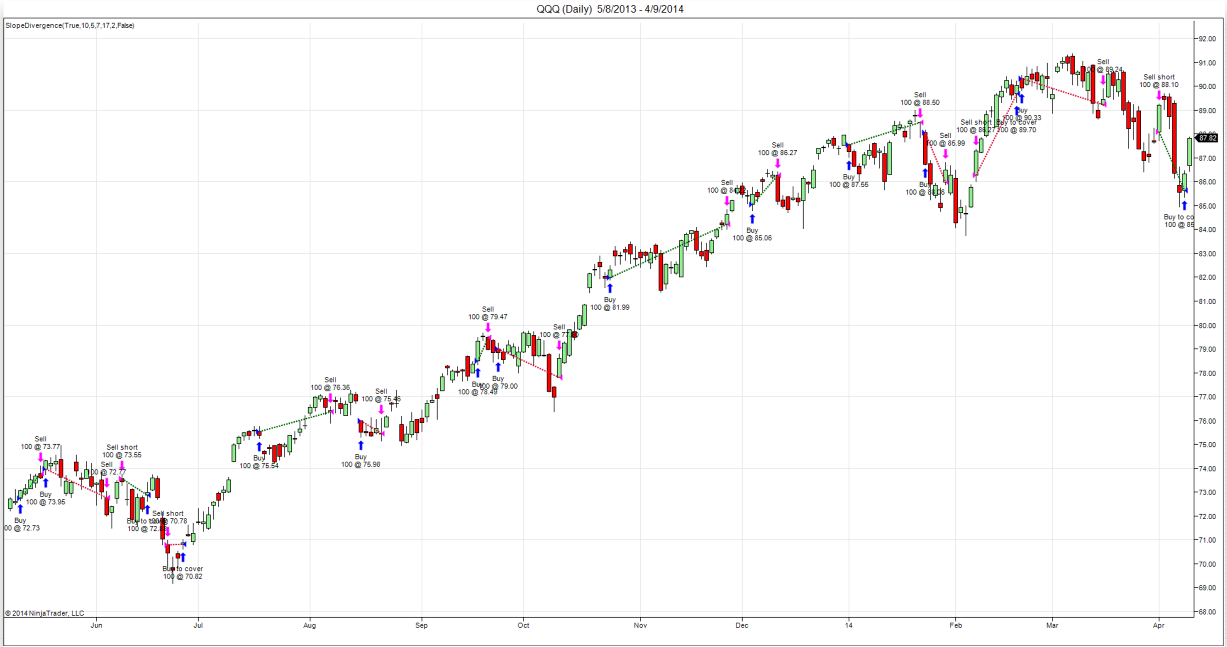


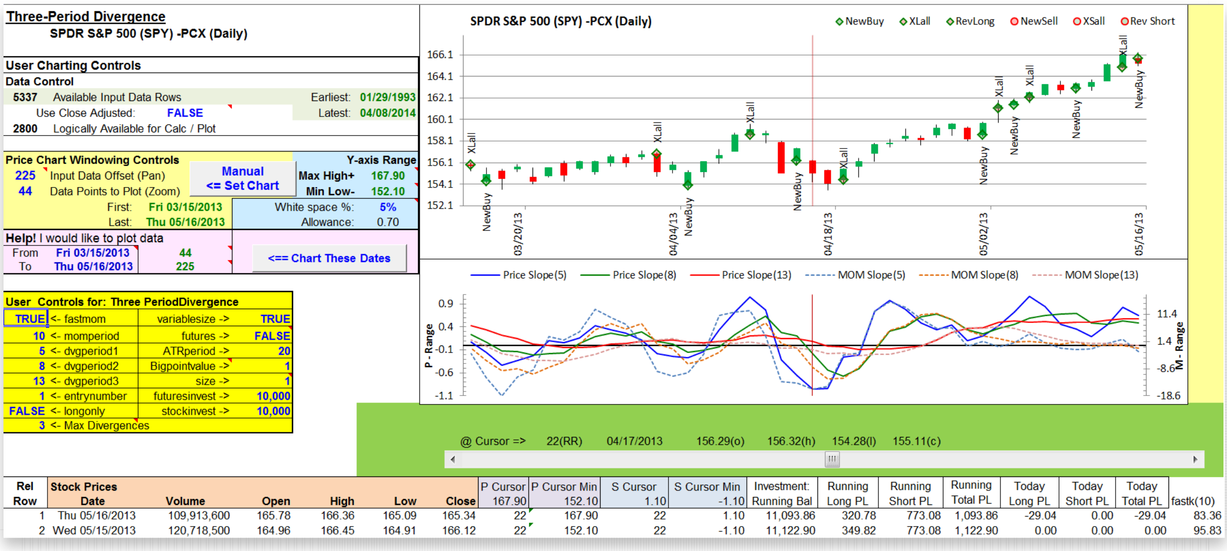
FIGURE 9: NINJATRADER. The screenshot shows the strategy applied to the QQQ daily chart in NinjaTrader.

—Raymond Deux, Cal Hueber and Jesse Nelson
NinjaTrader, LLC

MICROSOFT EXCEL: JUNE 2014

In “Slope Divergence: Capitalizing On Uncertainty” in this issue, author Perry Kaufman walks us through the use of divergence between price action and momentum to determine long and short entries & exits. Along the way, he shows us how to build a simple trading system based on these signals.

Figure 10 shows a chart that resembles Sidebar Figure 1 from Kaufman’s article. The indicator portion of the chart in his sidebar only shows price slopes calculated at three periods. I have enhanced (or cluttered?) the indicator part of the chart by including the momentum slopes.



Three-Period Divergence
SPDR S&P 500 (SPY) -PCX (Daily)

User Charting Controls
Data Control
5337 Available Input Data Rows — Earliest: 01/29/1993
Use Close Adjusted: FALSE — Latest: 04/08/2014
2800 Logically Available for Calc / Plot

Price Chart Windowing Controls
225 Input Data Offset (Pan)
44 Data Points to Plot (Zoom)
First: Fri 03/15/2013
Last: Thu 05/16/2013

Y-axis Range
Max High+ 167.90
Min Low- 152.10
White space %: 5%
Allowance: 0.70

Help! I would like to plot data
From Fri 03/15/2013 44
To Thu 05/16/2013 225

User Controls for: Three PeriodDivergence			
TRUE	<- fastmom	variablesize ->	TRUE
10	<- momperiod	futures ->	FALSE
5	<- dvgperiod1	ATRperiod ->	20
8	<- dvgperiod2	Bigpointvalue ->	1
13	<- dvgperiod3	size ->	1
1	<- entrynumber	futuresinvest ->	10,000
FALSE	<- longonly	stockinvest ->	10,000
3	<- Max Divergences		

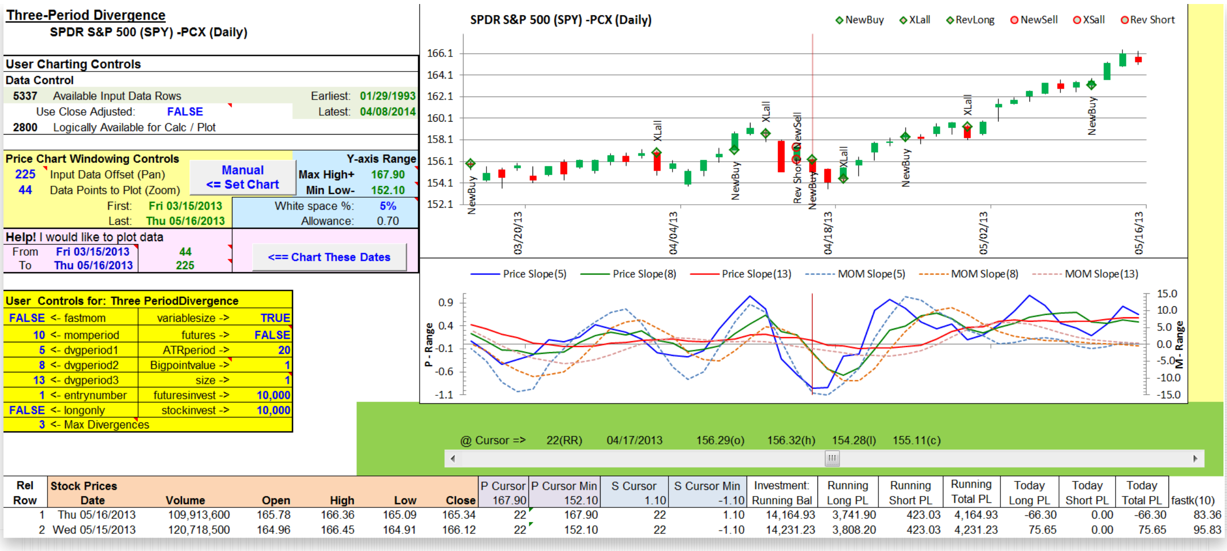
@ Cursor => 22(RR) 04/17/2013 156.29(o) 156.32(h) 154.28(l) 155.11(c)

Rel	Stock Prices						P Cursor	P Cursor Min	S Cursor	S Cursor Min	Investment:	Running	Running	Running	Today	Today	Today	
Row	Date	Volume	Open	High	Low	Close	167.90	152.10	1.10	-1.10	Running Bal	Long PL	Short PL	Total PL	Long PL	Short PL	Total PL	fastk(10)
1	Thu 05/16/2013	109,913,600	165.78	166.36	165.09	165.34	22	167.90	22	1.10	11,093.86	320.78	773.08	1,093.86	-29.04	0.00	-29.04	83.36
2	Wed 05/15/2013	120,718,500	164.96	166.45	164.91	166.12	22	152.10	22	-1.10	11,122.90	349.82	773.08	1,122.90	0.00	0.00	0.00	95.83

FIGURE 10: EXCEL, Trade Signals Using Fast Momentum=True

As you can see, the range of the price slopes and the range of the momentum slopes are radically different in scale. However, we do not care about this difference in scale as long as the horizontal axis is the zero line for both sets of indicators. The logic of this system makes decisions based on how many of these indicator lines and in which combinations are above or below the zero line at any given time.

Figure 11 shows the results of changing just one of the user controls. The zero crossings of the slope indicators have shifted. The divergence combinations have changed and so have our entries & exits.



Three-Period Divergence
SPDR S&P 500 (SPY) -PCX (Daily)

User Charting Controls

Data Control

5337	Available Input Data Rows	Earliest:	01/29/1993
	Use Close Adjusted: FALSE	Latest:	04/08/2014
2800	Logically Available for Calc / Plot		

Price Chart Windowing Controls			Y-axis Range	
225	Input Data Offset (Pan)	Manual <= Set Chart	Max High+	167.90
44	Data Points to Plot (Zoom)		Min Low-	152.10
	First:	Fri 03/15/2013	White space %:	5%
	Last:	Thu 05/16/2013	Allowance:	0.70

Help! I would like to plot data

From	Fri 03/15/2013	44	<== Chart These Dates
To	Thu 05/16/2013	225	

User Controls for: Three PeriodDivergence

FALSE	<- fastmom	variablesize ->	TRUE
10	<- momperiod	futures ->	FALSE
5	<- dvgperiod1	ATRperiod ->	20
8	<- dvgperiod2	Bigpointvalue ->	1
13	<- dvgperiod3	size ->	1
1	<- entrynumber	futuresinvest ->	10,000
FALSE	<- longonly	stockinvest ->	10,000
3	<- Max Divergences		

@ Cursor => 22(RR) 04/17/2013 156.29(o) 156.32(h) 154.28(l) 155.11(c)

Rel Row	Stock Prices Date	Volume	Open	High	Low	Close	P Cursor 167.90	P Cursor Min 152.10	S Cursor 1.10	S Cursor Min -1.10	Investment: Running Bal	Running Long PL	Running Short PL	Running Total PL	Today Long PL	Today Short PL	Today Total PL	fastk(10)
1	Thu 05/16/2013	109,913,600	165.78	166.36	165.09	165.34	22	167.90	22	1.10	14,164.93	3,741.90	423.03	4,164.93	-66.30	0.00	-66.30	83.36
2	Wed 05/15/2013	120,718,500	164.96	166.45	164.91	166.12	22	152.10	22	-1.10	14,231.23	3,808.20	423.03	4,231.23	75.65	0.00	75.65	95.83

FIGURE 11: EXCEL, Changing Only One Control. When we set fast momentum=false, it alters the divergence conditions and thus the signals.

In these two charts, we are only viewing the most recent 44 bars. But under the covers, the spreadsheet calculations include 2,800 bars: 4/4/2002 through 5/16/2013, or just over 11 years.

In Figure 10, we started on 4/4/2002 with a nest egg of $10,000 and grew it to $11,093.86 (see the *investment running balance* and *running total PL* columns). Contrast that with Figure 11, where we again started on 4/4/2002 with the same $10,000 and with one change to the controls, grew it to $14.164.93. Compare also the two plots of the *running balance* under each of the two settings in Figures 12 and 13.

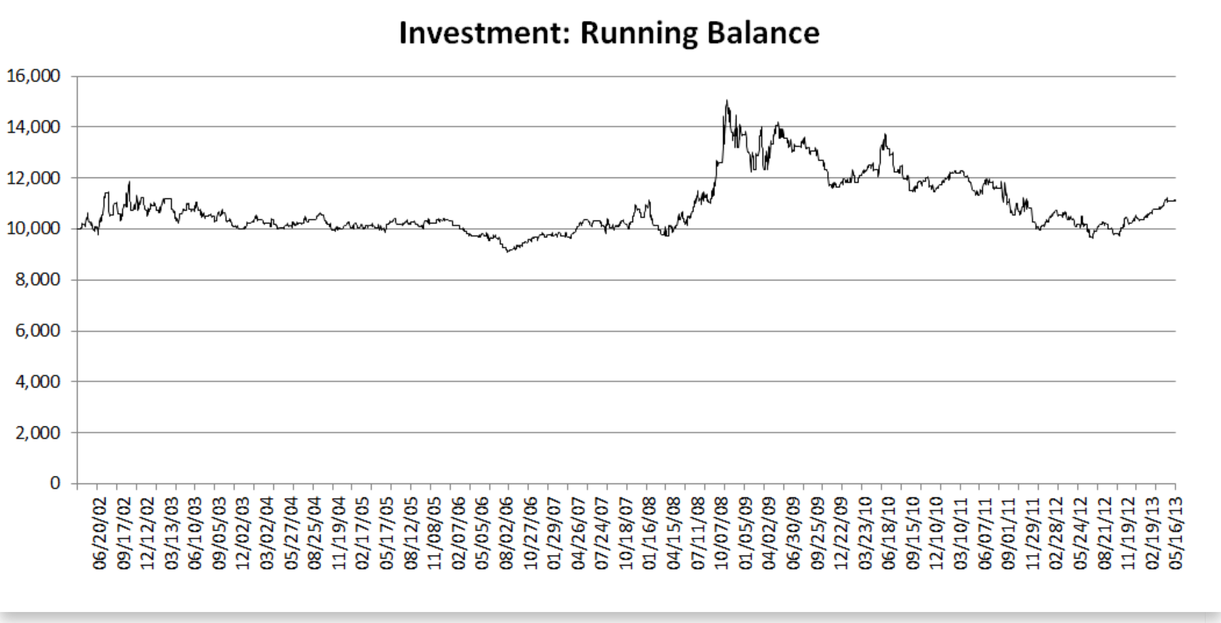


FIGURE 12: EXCEL, RUNNING BALANCE, Fast Momentum=True. Here are sample results for the running balance with the user controls as set in

Figure 10 (that is, fast momentum=true).

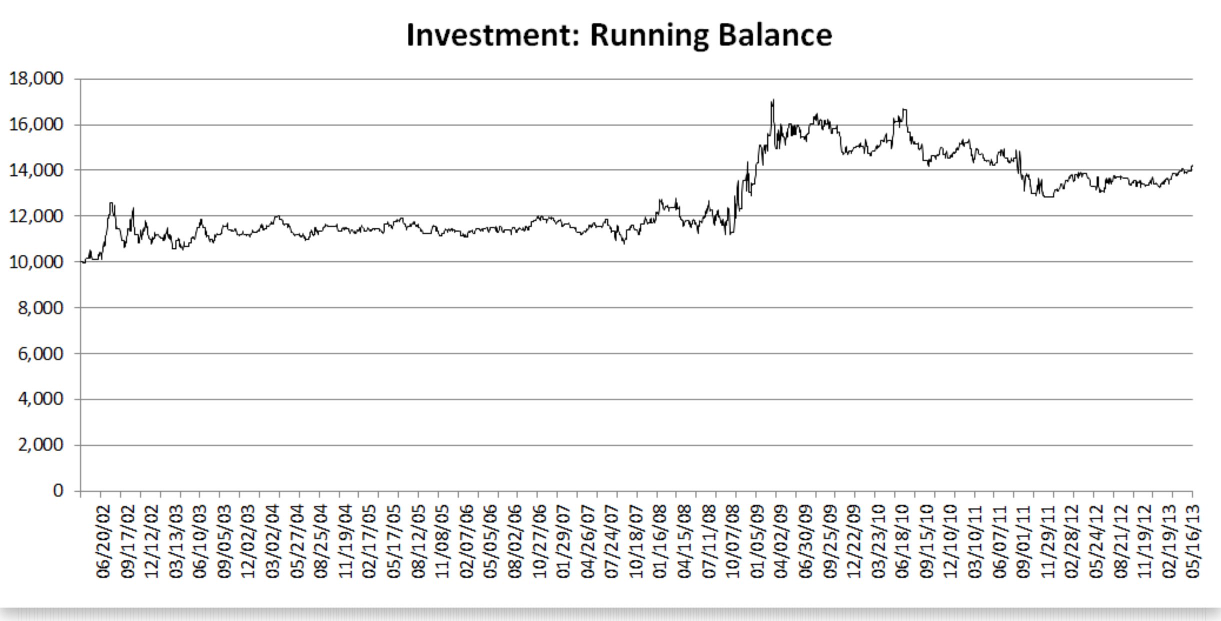


FIGURE 13: EXCEL, RUNNING BALANCE, Fast Momentum=FALSE. Here are sample results for the running balance with the user controls as set in Figure 11 (that is, fast momentum=false).

In his article, Kaufman presents us with a flexible system with eight controls that influence the trade decision logic. Here, we have explored one of the eight. Enjoy!

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" 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 June 2014 issue of
Technical Analysis of STOCKS & COMMODITIES magazine.

July 2014

For this month's Traders' Tips, the focus is mainly Sylvain Vervoort's article in this issue, "Exploring Charting Techniques, Part 1," with some Traders' Tips focusing on topics from recent issues instead. Here we present the July 2014 Traders' Tips code with possible implementations in various software.

Code for NinjaTrader is already provided in Vervoort's article. S&C subscribers will find that code at the Subscriber Area of our website. 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 are contributed by various software developers or programmers for software that is capable of customization.

TRADESTATION: JULY 2014

In "Exploring Charting Techniques, Part 1" in this issue, author Sylvain Vervoort presents an overview of some of the many bar types that traders can choose to use while making their trading decisions. Vervoort starts with simple line charts, which are nothing more than a line connecting points that represent some aspect of the market that changes. Each point is a line segment endpoint, and the offset between points is the chart interval. Vervoort states he doesn't find these useful, as too much data is ignored. Next, he reviews *open, high, low, close* (OHLC) representations: bar, tick bar, and candlestick charts. Both are able to display the OHLC data, with differing visual emphasis for the trader. The author states he feels these bar types present the trader with too much noise.

The author then reviews two bar types that might offer an answer to the noise issue: *range bars* and *renko bars*. He raises three issues with regard to autotrading and backtesting. First, he expresses that he doesn't like the lack of wicks. Next, he points out that the actual open and close may not be the same as those displayed on the chart. Finally, he states that gaps could be filled with untradable virtual bars.

All of these bar types discussed in Vervoort's article can be found in the TradeStation platform. The user can choose the type to use by formatting the symbol and selecting the desired bar types. As the author points out, advanced chart types can be automated only when the limitations are fully understood. More on strategy backtesting and automation of TradeStation's range, renko, and other advanced chart types can be found in the *advanced chart type* entry within *platform help*. (To reach this area: From the TradeStation platform help menu, select "platform help.")

For more information about EasyLanguage in general, see http://www.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 & Mark Mills
TradeStation Securities, Inc.
www.TradeStation.com

BACK TO LIST

eSIGNAL: JULY 2014

For this month's Traders' Tip, we've provided the formulas RelativeVolume.efs and FreedomOfMovement.efs based on the formulas described in Melvin Dickover's article in the April 2014 issue of S&C, "Evidence-Based Support & Resistance."

The 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 is shown in Figure 1.

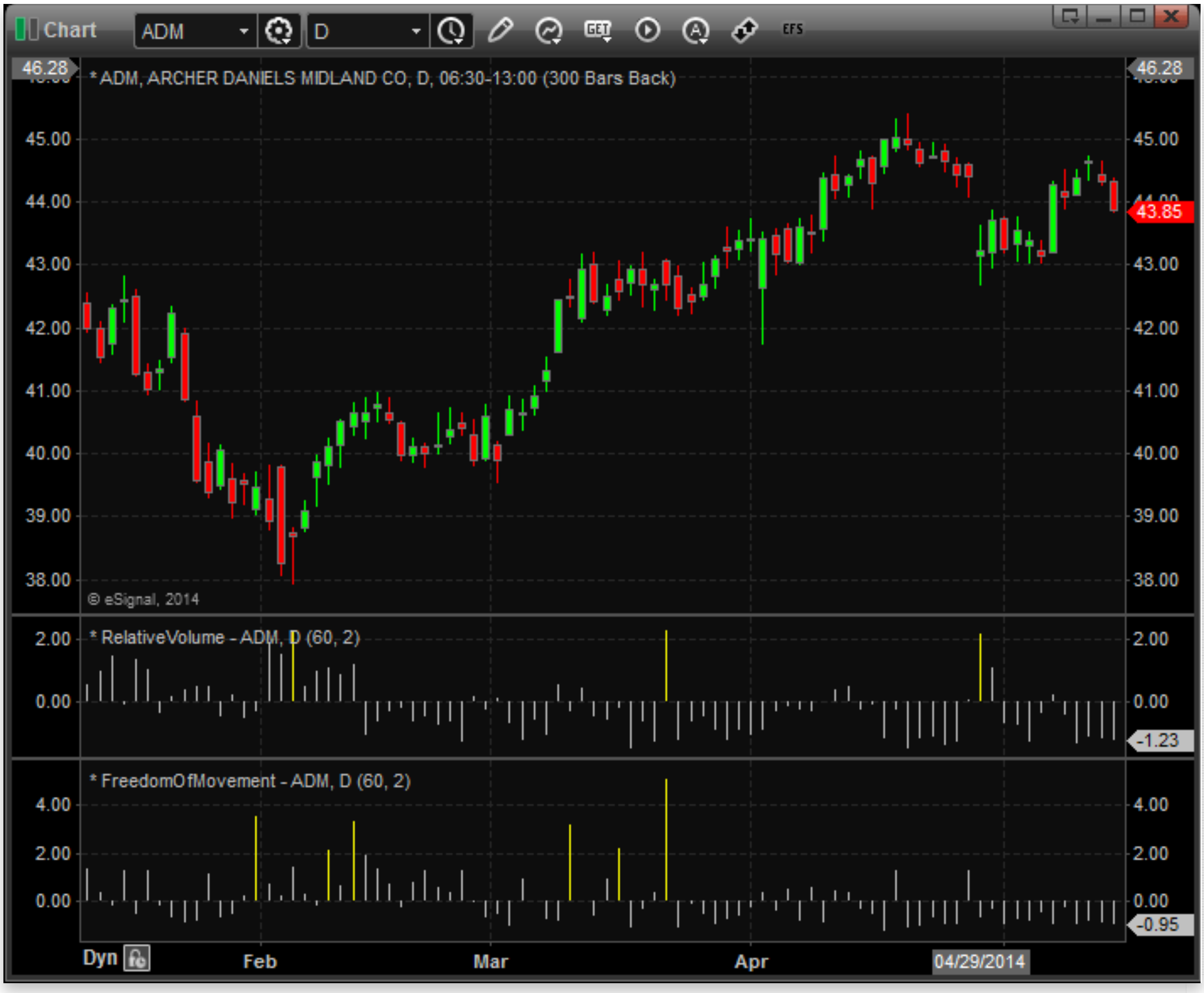


Chart
ADM
D
EFS
46.28
* ADM, ARCHER DANIELS MIDLAND CO, D, 06:30-13:00 (300 Bars Back)
45.00
44.00
43.85
43.00
42.00
41.00
40.00
39.00
38.00
© eSignal, 2014
* RelativeVolume - ADM, D (60, 2)
2.00
0.00
-1.23
* FreedomOfMovement - ADM, D (60, 2)
4.00
2.00
0.00
-0.95
Dyn
Feb
Mar
Apr
04/29/2014

FIGURE 1: eSIGNAL

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 scripts (EFS) are also available for copying & pasting below.

RelativeVolume.efs

```
/*********************************
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:
    RelativeVolume by Melvin E. Dickover:
    finds spikes of volume above numStDevs standard
    deviations of the average volume of the lookback period.

Formula Parameters:                        Default:
Period                                     60
```

```
StDevs                                  2

Version:            1.00  14/05/2014

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("RelativeVolume");
    setPlotType(PLOTTYPE_HISTOGRAM);

    var x = 0;

    fpArray[x] = new FunctionParameter("fpPeriod", FunctionParameter.NUMBER);
    with(fpArray[x++])
    {
        setName("Period");
        setDefault(60);
        setLowerLimit(1);
    }

    fpArray[x] = new FunctionParameter("fpNumStDevs", FunctionParameter.NUMBER);
    with(fpArray[x++])
    {
        setName("StDevs");
        setDefault(2);
    }
}

var bInit = false;
var bVersion = null;

var x_Volume = null;
var x_av = null;
var x_sd = null;
var x_relVol = null;

function main(fpPeriod, fpNumStDevs)
{
    if (bVersion == null) bVersion = verify();
    if (bVersion == false) return;

    if (!bInit)
    {
        x_Volume = volume();

        x_av = sma(fpPeriod, x_Volume);
        x_sd = efsInternal("Calc_Std", fpPeriod, x_Volume);
        x_relVol = efsInternal("Calc_Rel", x_Volume, x_av, x_sd);

        bInit = true;
    }

    var n_RelVol = x_relVol.getValue(0);
    if (n_RelVol == null)
        return;

    if (n_RelVol > fpNumStDevs)
        setBarFgColor(Color.blue)
    else
        setBarFgColor(Color.grey);

    return n_RelVol;
}

var xSMA = null;

function Calc_Std(nPeriod, xSourse)
{
    if (getBarState() == BARSTATE_ALLBARS)
    {
        xSMA = sma(nPeriod, xSourse);
    }

    var nSMA = xSMA.getValue(0);

    if (nSMA == null)
        return;

    var nSum = 0;

    for (var i = 0; i < nPeriod; i++)
    {
       var nSource = xSourse.getValue(-i);

       if (nSource == null)
           return;

       var nVal = Math.pow((nSource - nSMA), 2);

       nSum += nVal;
    }

    var nReturnValue = Math.sqrt(nSum / nPeriod);

    return nReturnValue;
}

function Calc_Rel(xSourse, xSourseAV, xSourseSD)
{
    var nSourse = xSourse.getValue(0);
    var nSourseAV = xSourseAV.getValue(0);
    var nSourseSD = xSourseSD.getValue(0);

    if(nSourse == null || nSourseAV == null || nSourseSD == null)
        return;

    var nReturnValue = (nSourse - nSourseAV) / nSourseSD;

    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;
}
```

FreedomOfMovement.efs

```
/*********************************
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:
    FreedomOfMovement by Melvin E. Dickover:
    computes how much effort to move the close up or down from previous bar.
    Effort is defined as the normalized relative volume.
    Effect is the normalized percent of (close - previous close).
    The resulting Freedom of Movement is the relative ratio of effort/effect.
    The larger the spike in FoM, the more the restriction of freedom of movement.
    Price movement is easy when FoM is small or negative,
    that is, only a small volume required to move price appreciably.

Formula Parameters:                     Default:
Period                                  60
StDevs                                  2

Version:            1.00  14/05/2014

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("FreedomOfMovement");
    setPlotType(PLOTTYPE_HISTOGRAM);

    var x = 0;

    fpArray[x] = new FunctionParameter("fpPeriod", FunctionParameter.NUMBER);
    with(fpArray[x++])
    {
        setName("Period");
        setDefault(60);
        setLowerLimit(1);
    }

    fpArray[x] = new FunctionParameter("fpNumStDevs", FunctionParameter.NUMBER);
    with(fpArray[x++])
    {
        setName("StDevs");
        setDefault(2);
    }
}

var bInit = false;
var bVersion = null;

var x_Volume = null;

var x_aMove = null;
var x_theMin = null;
var x_theMax = null;
var x_theMove = null;

var x_av = null;
var x_sd = null;
var x_relVol = null;

var x_theMinV = null;
var x_theMaxV = null;
var x_theVol = null;

var x_vByM = null;
var x_avF = null;
var x_sdF = null;
var x_theFoM = null;

function main(fpPeriod, fpNumStDevs)
{
    if (bVersion == null) bVersion = verify();
    if (bVersion == false) return;

    if (!bInit)
    {
        x_aMove = efsInternal("Calc_aMove");
        x_theMin = llv(fpPeriod, x_aMove);
        x_theMax = hhv(fpPeriod, x_aMove);
        x_theMove = efsInternal("Calc_Effort_Effect", x_aMove, x_theMin, x_theMax);

        x_Volume = volume();
        x_av = sma(fpPeriod, x_Volume);
        x_sd = efsInternal("Calc_Std", fpPeriod, x_Volume);
        x_relVol = efsInternal("Calc_Rel", x_Volume, x_av, x_sd);

        x_theMinV = llv(fpPeriod, x_relVol);
        x_theMaxV = hhv(fpPeriod, x_relVol);
        x_theVol = efsInternal("Calc_Effort_Effect", x_relVol, x_theMinV, x_theMaxV);

        x_vByM = efsInternal("Calc_vByM", x_theMove, x_theVol);

        x_avF = sma(fpPeriod, x_vByM);
        x_sdF = efsInternal("Calc_Std",fpPeriod, x_vByM);
        x_theFoM = efsInternal("Calc_Rel", x_vByM, x_avF, x_sdF);

        bInit = true;
    }

    var n_theFoM = x_theFoM.getValue(0);

    if (n_theFoM == null)
        return;

    if (n_theFoM < fpNumStDevs)
        setBarFgColor(Color.grey)
    else
        setBarFgColor(Color.blue);

    return n_theFoM;
}

var xSMA = null;

function Calc_Std(nPeriod, xSourse)
{
    if (getBarState() == BARSTATE_ALLBARS)
    {
        xSMA = sma(nPeriod, xSourse);
    }

    var nSMA = xSMA.getValue(0);

    if (nSMA == null)
        return;

    var nSum = 0;

    for (var i = 0; i < nPeriod; i++)
    {
       var nSource = xSourse.getValue(-i);

       if (nSource == null)
           return;

       var nVal = Math.pow((nSource - nSMA), 2);

       nSum += nVal;
    }

    var nReturnValue = Math.sqrt(nSum / nPeriod);

    return nReturnValue;
```

```
}

function Calc_Rel(xSourse, xSourseAV, xSourseSD)
{
    var nSourse = xSourse.getValue(0);
    var nSourseAV = xSourseAV.getValue(0);
    var nSourseSD = xSourseSD.getValue(0);

    if(nSourse == null || nSourseAV == null || nSourseSD == null)
        return;

    var nReturnValue = (nSourse - nSourseAV) / nSourseSD;

    return nReturnValue;
}

var xClose = null;

function Calc_aMove()
{
    if (getBarState() == BARSTATE_ALLBARS)
    {
        xClose = close();
    }

    var nClose = xClose.getValue(0);
    var nPrevClose = xClose.getValue(-1);

    if (nClose == null || nPrevClose == null)
        return;

    var nReturnValue = Math.abs((nClose - nPrevClose) / nPrevClose);

    return nReturnValue;
}

function Calc_Effort_Effect(xSourse, xSourceMin, xSourceMax)
{
    var nSourse = xSourse.getValue(0);
    var nSourseMin = xSourceMin.getValue(0);
    var nSourseMax = xSourceMax.getValue(0);

    if(nSourse == null || nSourseMin == null || nSourseMax == null)
        return;

    var nReturnValue = 0;

    if (nSourseMax > nSourseMin)
        nReturnValue = 1 + ((nSourse - nSourseMin) * (10 - 1)) / (nSourseMax - 
nSourseMin);

    return nReturnValue;
}

function Calc_vByM(xSourceMove, xSourceVol)
{
    var nSourceMove = xSourceMove.getValue(0);
    var nSourceVol = xSourceVol.getValue(0);

    if(nSourceMove == null || nSourceVol == null)
```

```
        return;

    return nSourceVol / nSourceMove;
}

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: JULY 2014

In Wealth-Lab, the renko charting style can be applied to the current chart from the *Function* toolbar → *ChartStyle*. To have the chart approximate the modified renko style as presented in “Exploring Charting Techniques, Part 1” by Sylvain Vervoort in this issue, just strike ctrl+Y (or choose *chart style settings* from the chart’s right-click context menu) and enable “overlay HLC chart.” The way the Wealth-Lab backtesting engine is designed, trades at market price occur at the opening price and the correct market prices are always reflected for trade execution when working with renko bricks.

As an example of applying a trading technique to the modified renko charts, Wealth-Lab users can download the “renko basic [Rev A]” strategy and switch the chart to HLC view as instructed. Downloading a public strategy is as easy as invoking the *open strategy* dialog (ctrl+O), clicking the *download* button, making sure you’ll get all available items by unchecking “Published since...”, and clicking *begin download*.

A sample chart is shown in Figure 2.

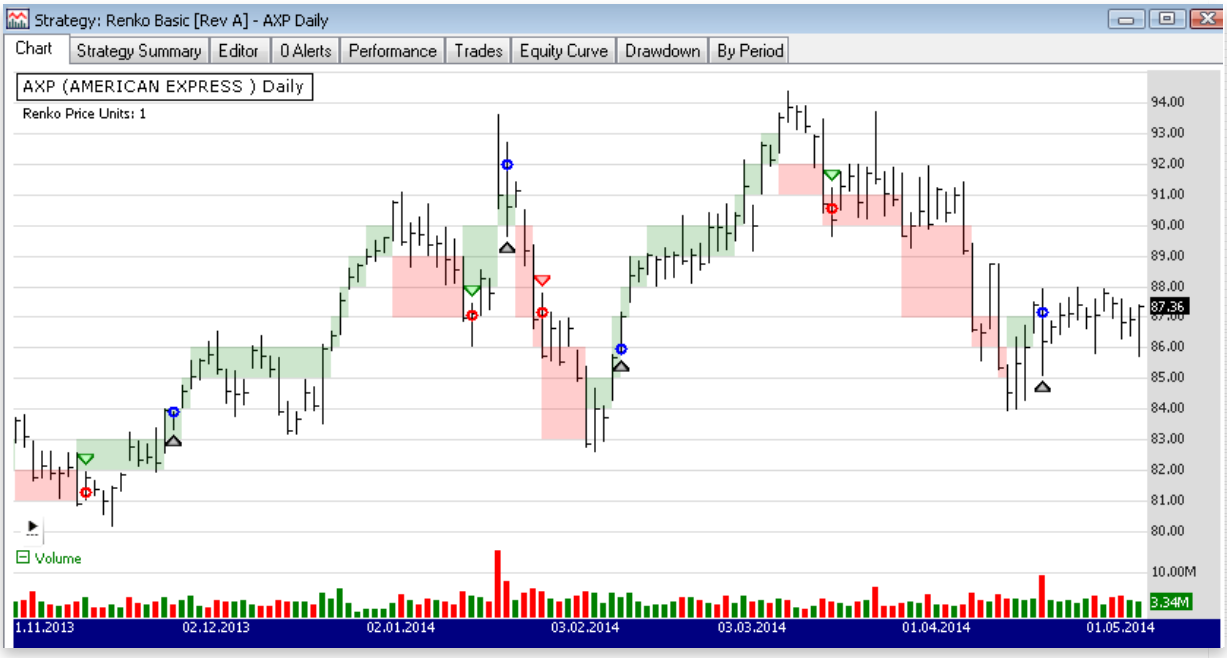


FIGURE 2: WEALTH-LAB, RENKO CHART. This sample daily chart of American Express (AXP) shows a tradable renko chart.

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

BACK TO LIST

AMIBROKER: JULY 2014

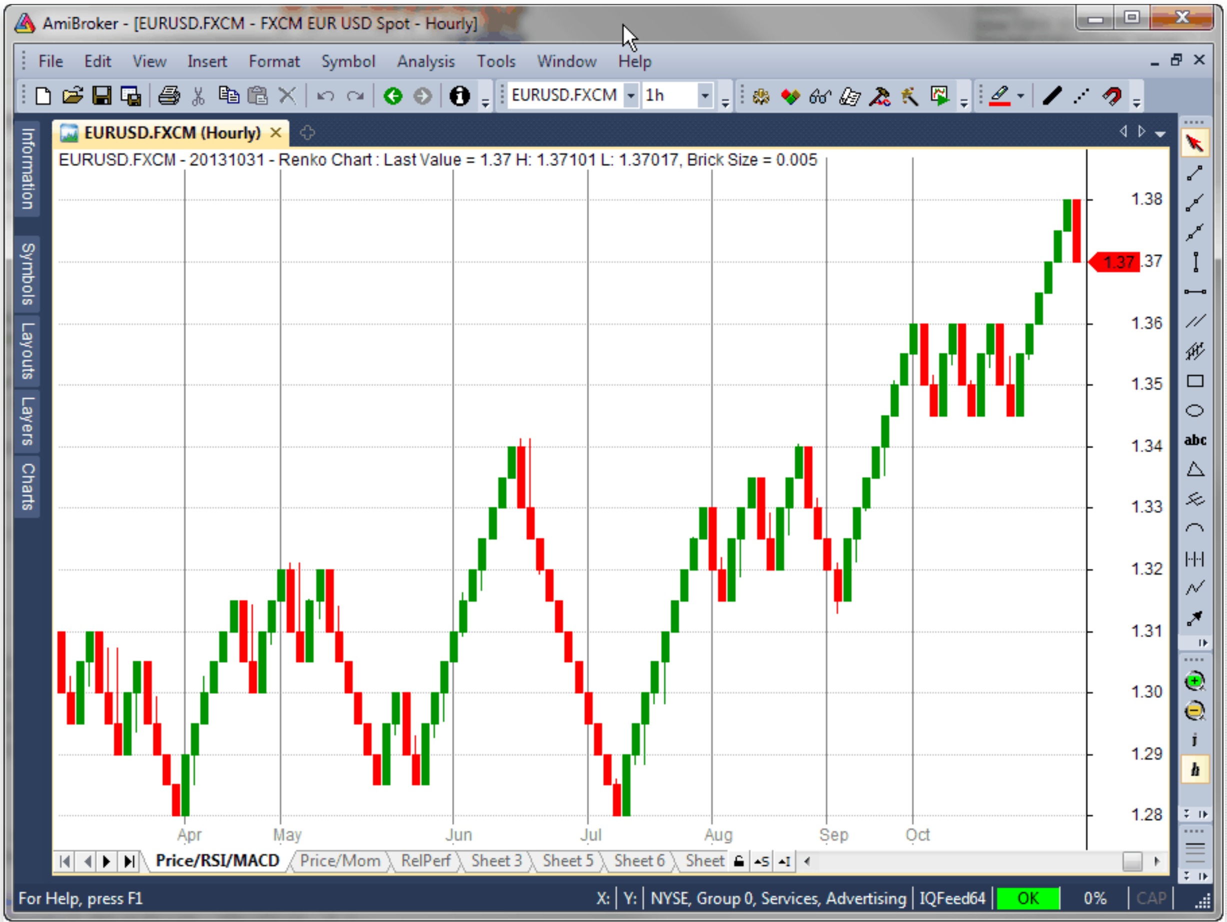
In "Exploring Charting Techniques, Part 1" in this issue, author Sylvain Vervoort presents modified renko charts that include high/low wicks and that reflect the correct open price for renko reversal points.

A ready-to-use formula for AmiBroker is shown below, based on Vervoort's article. You can select between *normal* or *modified* renko charts using the parameters window.

```
// Modified Renko Chart with custom date axis
// and high/low winks
// Loosely based on Renko chart formula by G. Kavanagh
// from AmiBroker on-line formula library (id=521)
// Modifications & fixes TJ 2014
```

```
function FillRun( dir, num, changedir )
{
   global i, j, modified, dt, RKC, RKO, RKD, RKH, RKL;

   for ( x = 1; x <= num AND j < BarCount - 1; x++ )
   {
     j++;

     extra = ( changedir AND x == 1 ) * dir;

     RKC[ j ] = RKC[ j - 1 ] + dir + extra;
     RKO[ j ] = RKC[ j - 1 ] + IIf( modified, 0, extra );
     RKD[ j ] = dt[ i ];
     RKH[ j ] = High[ i - 1 ];
     RKL[ j ] = Low[ i - 1 ];
   }
}

SetBarsRequired( sbrAll, sbrAll );
Brick = Param( "Brick Size", 0.001, 0.0001, 1.00, 0.001 );
reverse = 2;
intra = ParamToggle( "Intraday", "No|Yes", 0 );
modified = ParamToggle( "Modified", "No|Yes", 0 );

// Convert the closing price to rising and falling rounded bricks
CF = ceil( C / Brick );
CR = floor( C / Brick );

// initialize first element
j = 0;
RKC[j] = CF[0];
RKO[j] = CF[0] + 1;
RKD = 0;
RKH = 0;
RKL = 0;
dt = IIf( intra, floor( TimeNum() / 100 ), DateNum() );

dir = -1; // 1 is up, -1 is down

// Loop to produce the Renko values in number of bricks

for ( i = 1; i < BarCount - 1; i++ )
{
 if ( j >= BarCount ) break; // no more room -> finish

 if ( CF[i] <= RKC[j] - 1 AND dir < 0 ) // Continue down
 {
    num = RKC[j] - CF[i];
    FillRun( dir, num, False );
 }
 else
 if ( CR[i] >= RKC[j] + Reverse AND dir < 0 ) // Change down to up
 {
    num = CR[i] - RKC[j];
    dir = 1;
    FillRun( dir, num, True );
 }
 else
 if ( CR[i] >= RKC[j] + 1 AND dir > 0 ) // Continue Up
 {
    num = CR[i] - RKC[j];
    FillRun( dir, num, False );
 }
 else
 if ( CF[i] <= RKC[j] - Reverse AND dir > 0 ) // Change up to down
 {
    num = RKC[j] - CF[i];
    dir = -1;
    FillRun( dir, num, True );
 }
}

// move the chart to right end of chart space, ie last brick on last bar position
delta =  BarCount - 1 - j;

RKC = Ref( RKC, -delta );
RKO = Ref( RKO, -delta );
RKD = Ref( RKD, -delta );
RKH = Ref( RKH, -delta );
RKL = Ref( RKL, -delta );

C = RKC * Brick;
O = RKO * Brick;
H = IIf( modified, RKH, Max( C, O ) );
L = IIf( modified, RKL, Min( C, O ) );

Plot( C, "", IIf( C > O, colorGreen, colorRed ), styleCandle );

xnum = floor( RKD / 100 );
XChange = IIf( xnum != Ref( xnum, -1 ), 1, Null );

Plot( XChange, "", colorGrey50, styleHistogram | styleOwnScale, 0, 1 );

// Draw renko-date axis
MonthNames = "Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec";
fvb = FirstVisibleValue( BarIndex() );
lvb = LastVisibleValue( BarIndex() );

for ( i = fvb; i < lvb; i++ )
{
  if ( XChange[ i ] )
  {
     if ( intra )
         datetext = StrFormat( "%02gh", floor ( RKD[ i ] / 100 ) );
     else
     if ( ( xnum[ i ] % 100 ) == 1 )
         datetext = StrFormat( "%04.0f", 1900 + ( xnum[ i ] / 100 ) );
     else
         datetext = StrExtract( MonthNames, ( xnum[ i ] % 100 ) - 1 );

     PlotText( datetext  , i, LowestVisibleValue( Low ),
colorGrey50,                       colorWhite, -20 );
  }
}

Title = Name() + StrFormat( " - 20%06.0f", RKD % 1000000 ) + " - Renko Chart : Last
Value = " + RKC * Brick + " H: " + RKH + " L: " + RKL + ", Brick Size = " + Brick;

GraphXSpace = 5;
```

A sample chart is shown in Figure 3.



EURUSD.FXCM - 20131031 - Renko Chart : Last Value = 1.37 H: 1.37101 L: 1.37017, Brick Size = 0.005

FIGURE 3: AMIBROKER, RENKO CHART. Here is a sample chart showing a modified renko chart of EUR/USD.

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

BACK TO LIST

NEUROSHELL TRADER: JULY 2014

The renko bar charts described by Sylvain Vervoort in his article in this issue, “Exploring Charting Techniques, Part 1,” may be implemented in NeuroShell Trader through the use of the *InterChart Tools Renko* add-in. The *InterChart Tools Renko bars* are virtual and perform their calculations using the same methods as traditional renko bars, but once a trading signal is generated by the renko bar, both the trade and fill are correctly displayed on the open of the next bar of the base chart.

To replicate the sample chart shown in Vervoort's article of the S&P futures, simply select *new indicator* from the Insert menu and enter the following in the appropriate locations of the indicator parameters:

```
IctRenko High (0.25, 1, 1, 3, high, low, volume)
                (both dots and lines displayed on the chart)
IctRenko Low (0.25, 1, 1, 3, high, low, volume)
                (both dots and lines displayed on the chart)
IctRenko Volume (0.25, 1, 1, 3, high, low, volume)
IctRenko Streak Counter (0.25, 1, 1, 3, high, low, volume)
```

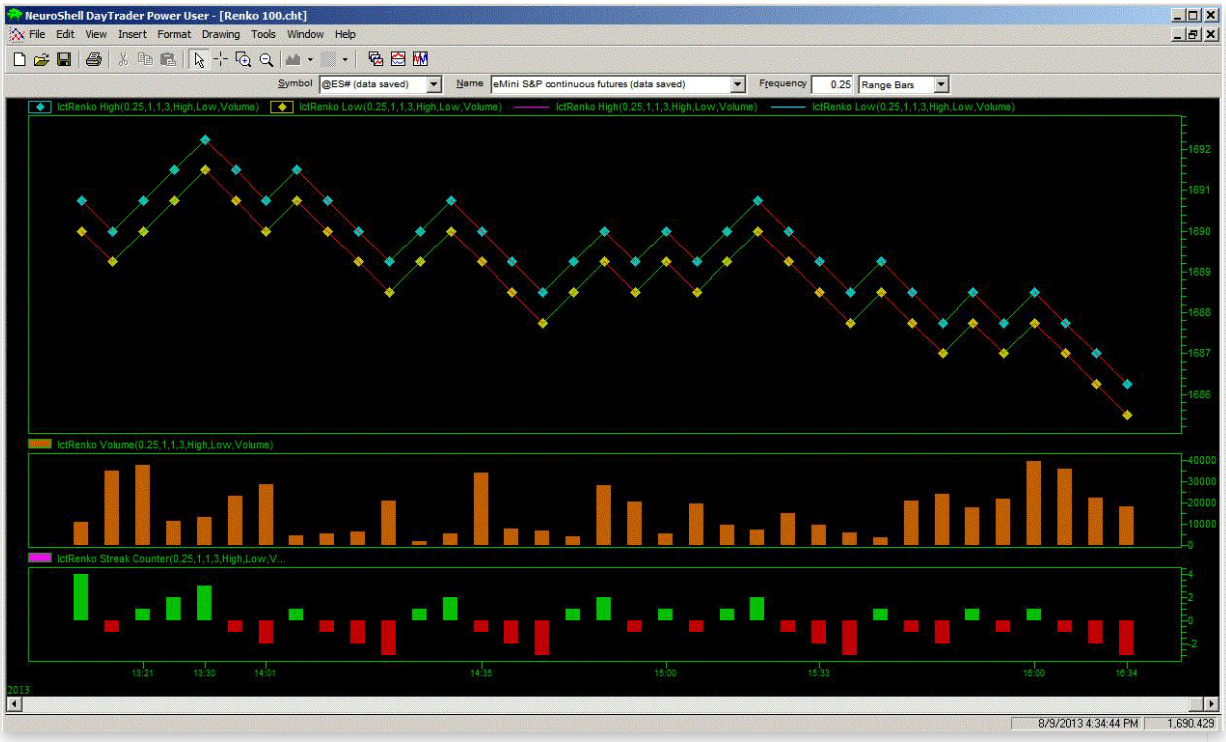
The value *0.25* corresponds to the size of the base chart's range bar, which is used to create the virtual renko bars. It is the virtual (or actual) tick size. You will need to specify a size that will not be overcome by noise for the security that is being traded. In the example shown, it is the actual tick size. The next two parameters represent the number of ticks used to calculate the up part of the renko bar, followed by the number of ticks used to compute the down part. The "3" represents a multiplier that is applied to the described renko bar's up/down ratio to realize its final size. This enables the indicators to use a different number of ticks for the up and down side of the renko bars. Since any bar's function is to absorb noise, and rising price jitter is often different from falling price jitter, our renko bars permit an asymmetrical definition to accommodate this.

The *renko volume* indicator displays the correct volume on the base chart that corresponds to the renko bars. The *streak counter* indicator displays the number of consecutive up or down renko bars that appear in sequence on the chart.

NeuroShell renko bars may, at the user's discretion, be controlled by the optimizer to identify the optimal bar size and noise absorption for a given algorithm or equity.

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.



NeuroShell DayTrader Power User - [Renko 100.cht]

Symbol: @ES# (data saved)
Name: eMini S&P continuous futures (data saved)
Frequency: 0.25 Range Bars

File Edit View Insert Format Drawing Tools Window Help
IctRenko High(0.25,1,1,3,High,Low,Volume)
IctRenko Low(0.25,1,1,3,High,Low,Volume)
IctRenko High(0.25,1,1,3,High,Low,Volume)
IctRenko Low(0.25,1,1,3,High,Low,Volume)
1692
1691
1690
1689
1688
1687
1686
IctRenko Volume(0.25,1,1,3,High,Low,Volume)
40000
30000
20000
10000
0
IctRenko Streak Counter(0.25,1,1,3,High,Low,V...
4
2
0
-2
13:21
13:30
14:01
14:36
15:00
15:33
16:00
16:34
2013
8/9/2013 4:34:44 PM
1,690.429

FIGURE 4: NEUROSHELL TRADER, RENKO BARS. **This NeuroShell Trader chart shows InterChart Tools renko bars.**

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

BACK TO LIST

AIQ: JULY 2014

The AIQ code I am providing this month is based on Mike Siroky's article from the May 2014 issue of STOCKS & COMMODITIES, "Wilder's RSI: Extending the Time Horizon." The code is provided at www.TradersEdgeSystems.com/traderstips.htm and is also shown at the end of this tip.

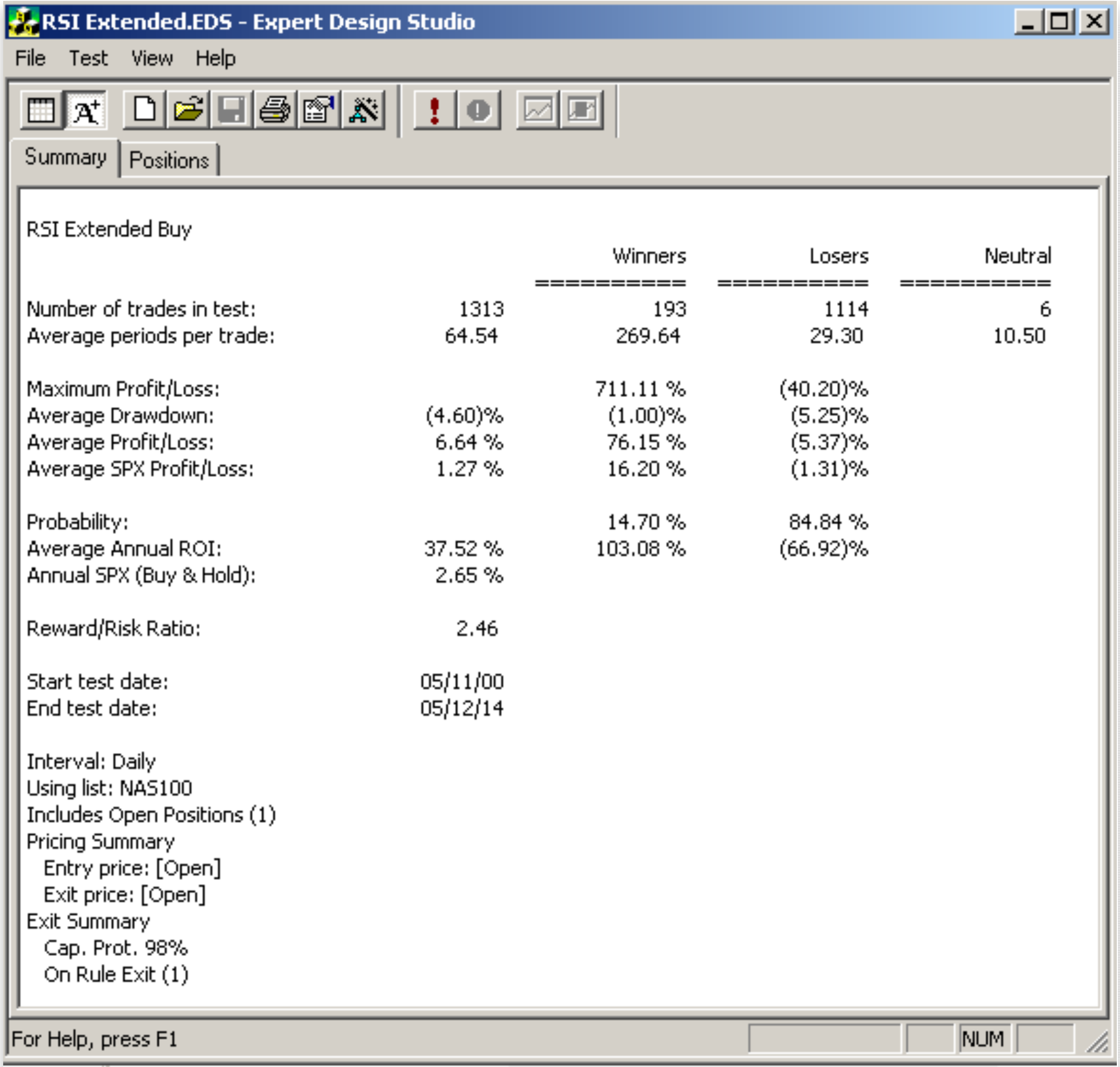
I am providing a system that uses Siroky's adjustable RSI bands that automatically adjust to the appropriate level for the input RSI length. The system is very simple:

- Buy the next bar at the market open when the RSI is less than the lower confidence interval band (RSI_CILOW)

- Exit the long position on the next bar at the market open when the RSI is greater than the upper confidence interval band (RSI_CIUP)
- For shorting, reverse these rules.

I have a parameter that allows testing long-only, short-only, or both long and short. The system lost when the short side was allowed to trade.

Figure 5 shows the AIQ EDS summary long-only backtest report using the NASDAQ 100 list of stocks over the period 5/11/2000 to 5/12/2014. Neither commission nor slippage has been subtracted from these results. In running this test, I used a capital protection of 98%, which is equivalent to a 2% stop-loss using the close. All entries and exits are at the next open. I could not get the short side to show a profit even with added market timing filters for trend on the NASDAQ 100 index.



```
RSI Extended Buy
                                              Winners        Losers        Neutral
                                           ==========    ==========     ==========
Number of trades in test:          1313           193          1114              6
Average periods per trade:        64.54        269.64         29.30          10.50

Maximum Profit/Loss:                         711.11 %      (40.20)%
Average Drawdown:               (4.60)%      (1.00)%        (5.25)%
Average Profit/Loss:             6.64 %       76.15 %       (5.37)%
Average SPX Profit/Loss:         1.27 %       16.20 %       (1.31)%

Probability:                                  14.70 %       84.84 %
Average Annual ROI:             37.52 %      103.08 %      (66.92)%
Annual SPX (Buy & Hold):         2.65 %

Reward/Risk Ratio:                2.46

Start test date:              05/11/00
End test date:                05/12/14

Interval: Daily
Using list: NAS100
Includes Open Positions (1)
Pricing Summary
  Entry price: [Open]
  Exit price: [Open]
Exit Summary
  Cap. Prot. 98%
  On Rule Exit (1)
```

FIGURE 5: AIQ. Here is the AIQ EDS summary long-only backtest report using the NASDAQ 100 list

of stocks over the period 5/11/2000 to 5/12/2014.

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 2014

The TradersStudio code I am providing this month is based on Mike Siroky's article from the May 2014 issue of STOCKS & COMMODITIES, "Wilder's RSI: Extending the Time Horizon." The code is provided at the following websites:

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

The following files are provided in the download:

- **System RSI_EXTENDED:** A trading system I created using the adjustable trading bands for the RSI that were suggested by Siroky in his article.
- **Function RSI_CILOW:** Returns the lower RSI trading band based on the inputs *RSI length* and the number of standard deviations that determine the confidence interval.
- **Function RSI_CIUP:** Returns the upper RSI trading band based on the inputs *RSI length* and the number of standard deviations that determine the confidence interval.

The system I am providing uses the author's adjustable RSI bands that automatically adjust to the appropriate level for the input RSI length. The system is very simple. The rules are:

- Buy the next bar at the market open when the RSI is less than the lower confidence interval band (RSI_CILOW).
- Exit the long position next bar at market open when the RSI is greater than the upper confidence interval band (RSI_CIUP).
- Reverse these rules for shorting.

I have a parameter that allows testing long-only, short-only, or both long and short. The system lost when the short side was allowed to trade.

I optimized the parameters on a sample of seven of the NASDAQ 100 stocks, and then I ran a test using all 100 of the NASDAQ 100 stocks. The equity curve trading 200 shares each of the

NASDAQ 100 list of stocks, trading long only, is shown in Figure 6 together with the underwater equity curve. The drawdown during bear markets might be improved by adding a market timing index trend filter.

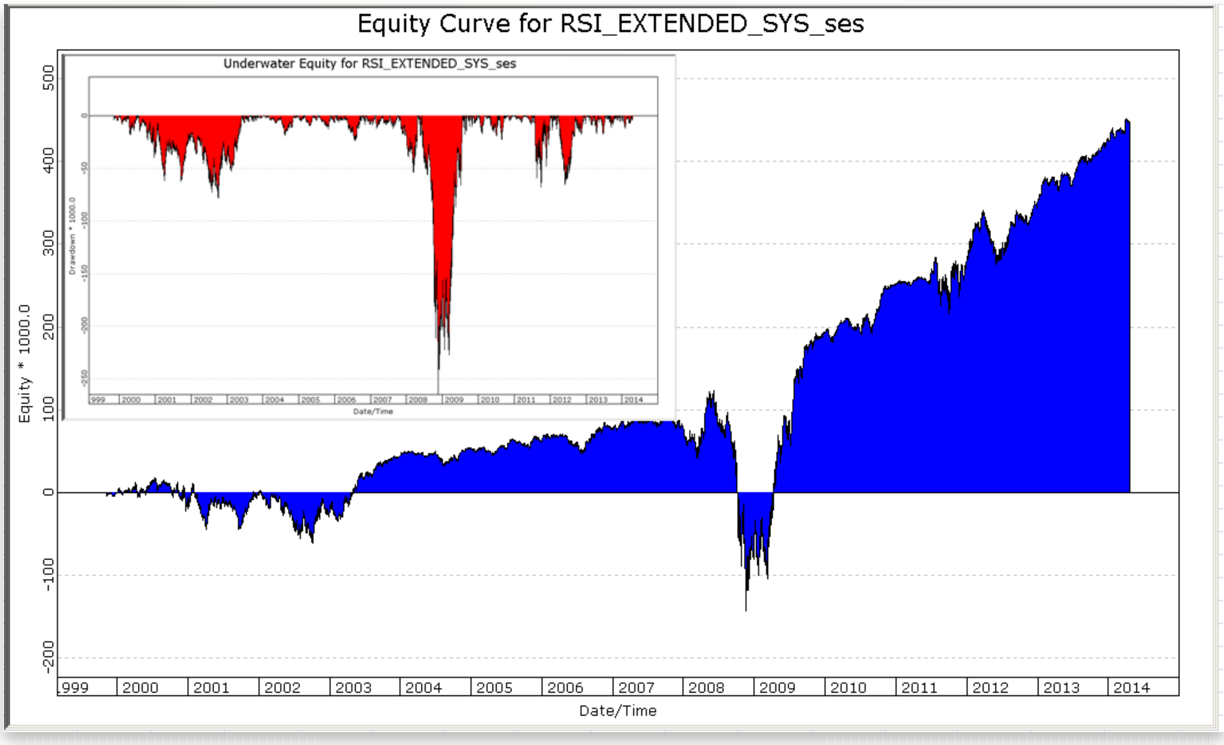


FIGURE 6: TRADERSSTUDIO, SAMPLE EQUITY CURVE. Here are sample equity and underwater equity curves trading the NASDAQ 100 list of stocks, long only, for the period 2000 through April 2014.

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

MICROSOFT EXCEL: JULY 2014

In "Exploring Charting Techniques, Part 1" in this issue, author Sylvain Vervoort presents his modified version of the renko bar construction method that is designed to support trend analysis in the presence of noisy-looking pricing data.

Since free, short-interval tick data for forex and other instruments is not generally available for more than one or two days back in time, I have not attempted to replicate the example charts from Vervoort's article, which used data from October 2013.

Fortunately, we can apply the SVERenkoBars logic to end-of-day data, as I have done here. Figures 7 and 8 give us a good representation of just how effective this technique is at clearing some of the visual price noise. The price shapes align fairly well; the reported time stamps, not so much.

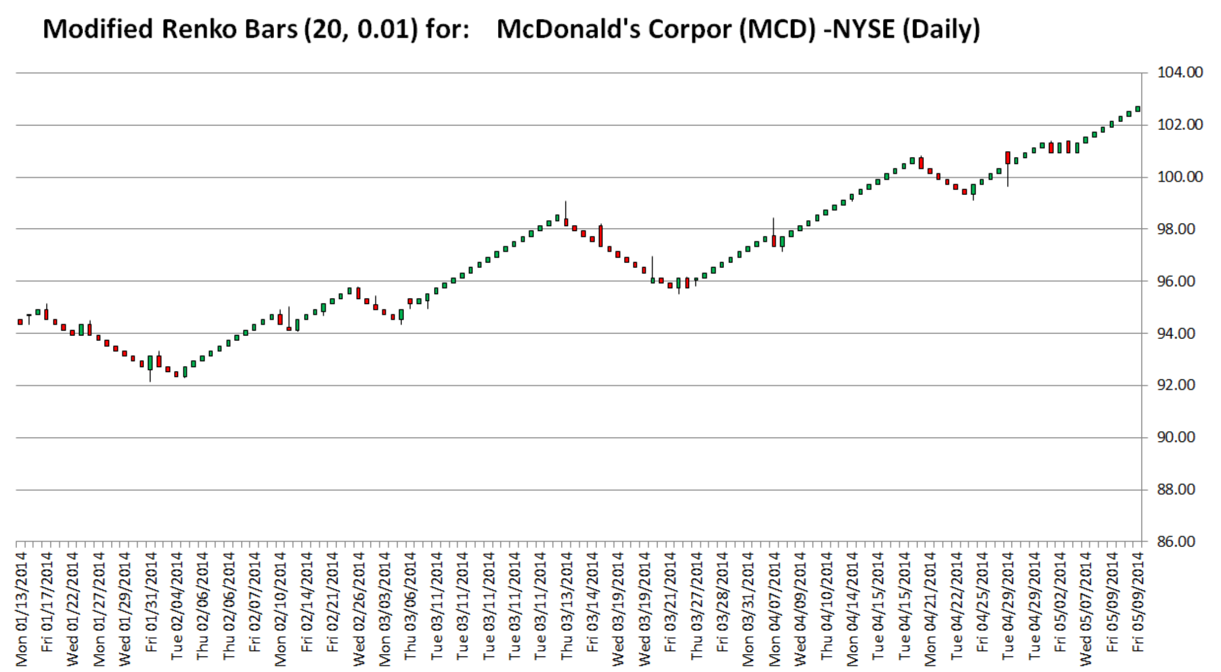


FIGURE 7: EXCEL, MODIFIED RENKO BARS. This sample chart shows modified renko bars based on Sylvain Vervoort's technique using daily data (end-of-day [EOD] ticks). See the SVERenkoBarsChart tab in the spreadsheet.

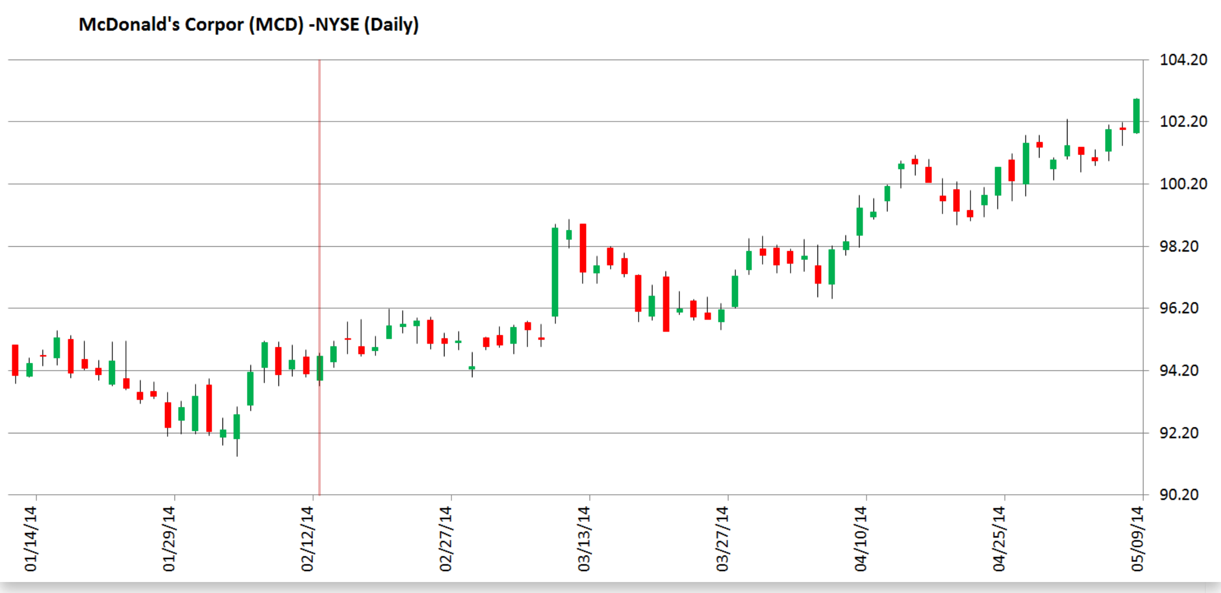
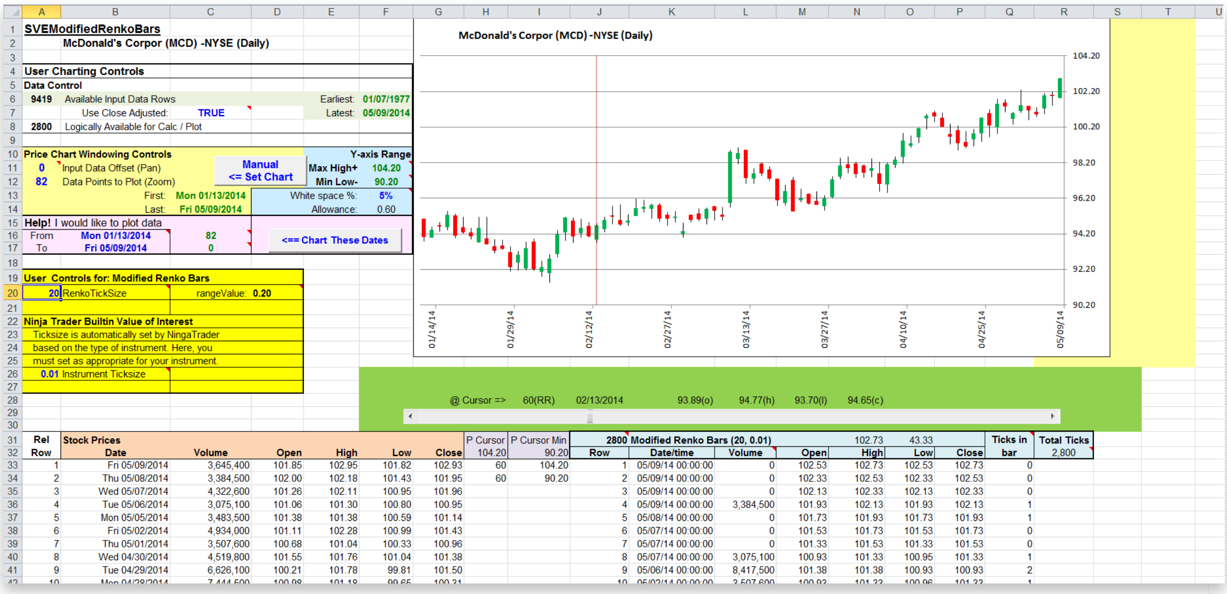


FIGURE 8: EXCEL, DAILY PRICE CANDLES. This sample chart shows daily price candles ("ticks") for approximately the same time period as the renko bars shown in Figure 7.

Generating Excel cell formulas to build renko bars was going to take far too long and make for a very convoluted spreadsheet. So I opted to create a user-defined array function (UDF) with VBA to build the bars.

Output from this array function occupies cell range J33:Q2832. That is the area with headers in light blue titled “modified renko bars” in Figure 9.

The advantage of the array function approach is that a large amount of data can be passed to a single execution of the function, and that single execution can return a lot of processed data to a large area of the spreadsheet. For this Traders’ Tip, the computation is very fast. And, as you can see in Figure 9, in terms of columns of formulas and values, this is the least-cluttered-looking spreadsheet to date.



SVEModifiedRenkoBars
McDonald's Corpor (MCD) -NYSE (Daily)

User Charting Controls
Data Control
9419 Available Input Data Rows — Earliest: 01/07/1977
Use Close Adjusted: TRUE — Latest: 05/09/2014
2800 Logically Available for Calc / Plot

Price Chart Windowing Controls — Y-axis Range
0 Input Data Offset (Pan) — Max High+ 104.20
82 Data Points to Plot (Zoom) — Min Low- 90.20
First: Mon 01/13/2014 — White space %: 5%
Last: Fri 05/09/2014 — Allowance: 0.60

Help! I would like to plot data
From Mon 01/13/2014 82
To Fri 05/09/2014 0

User Controls for: Modified Renko Bars
20 RenkoTickSize — rangeValue: 0.20

Ninja Trader Builtin Value of Interest
Ticksize is automatically set by NingaTrader
based on the type of instrument. Here, you
must set as appropriate for your instrument.
0.01 Instrument Ticksize

@ Cursor => 60(RR) 02/13/2014 93.89(o) 94.77(h) 93.70(l) 94.65(c)

Rel	Stock Prices						P Cursor	P Cursor Min	2800 Modified Renko Bars (20, 0.01)				102.73	43.33		Ticks in	Total Ticks
Row	Date	Volume	Open	High	Low	Close	104.20	90.20	Row	Date/time	Volume	Open	High	Low	Close	bar	2,800
1	Fri 05/09/2014	3,645,400	101.85	102.95	101.82	102.93	60	104.20	1	05/09/14 00:00:00	0	102.53	102.73	102.53	102.73	0	
2	Thu 05/08/2014	3,384,500	102.00	102.18	101.43	101.95	60	90.20	2	05/09/14 00:00:00	0	102.33	102.53	102.33	102.53	0	
3	Wed 05/07/2014	4,322,600	101.26	102.11	100.95	101.96			3	05/09/14 00:00:00	0	102.13	102.33	102.13	102.33	0	
4	Tue 05/06/2014	3,075,100	101.06	101.30	100.80	100.95			4	05/09/14 00:00:00	3,384,500	101.93	102.13	101.93	102.13	1	
5	Mon 05/05/2014	3,483,500	101.38	101.38	100.59	101.14			5	05/08/14 00:00:00	0	101.73	101.93	101.73	101.93	1	
6	Fri 05/02/2014	4,934,000	101.11	102.28	100.99	101.43			6	05/07/14 00:00:00	0	101.53	101.73	101.53	101.73	0	
7	Thu 05/01/2014	3,507,600	100.68	101.04	100.33	100.96			7	05/07/14 00:00:00	0	101.33	101.53	101.33	101.53	0	
8	Wed 04/30/2014	4,519,800	101.55	101.76	101.04	101.38			8	05/07/14 00:00:00	3,075,100	100.93	101.33	100.95	101.33	1	
9	Tue 04/29/2014	6,626,100	100.21	101.78	99.81	101.50			9	05/06/14 00:00:00	8,417,500	101.38	101.38	100.93	100.93	2	
10	Mon 04/28/2014	7,444,500	100.98	101.18	99.65	100.31			10	05/02/14 00:00:00	3,507,600	100.93	101.33	100.96	101.33	1	

FIGURE 9: EXCEL, NO CLUTTER. This is easily the least-cluttered-looking spreadsheet.

Inside a user-defined function, the VBA code can be every bit as procedural and complex as, for example, the NinjaTrader code that Vervoort provides in his article in this issue. In fact, the main line of the VBA code for this user-defined function looks very much like the main line of Vervoort’s NinjaTrader code.

In the background are two user-defined VBA class modules, one class to represent the bars and one to make functional routines available to our UDF mainline that is designed to approximate the names and actions of the bar creation, storage, and manipulation functions used in Vervoort’s example code.

As to renko bar user controls: There are only two on this spreadsheet. The *RenkoTickSize* value, which is the number of “instrument” ticks per renko brick, and the *instrument tick size*. The instrument tick size is a NinjaTrader built-in feature that is automatically set in NinjaTrader based on the type of instrument the user is analyzing at any given time. Since Excel is not NinjaTrader, the spreadsheet user must set this stand-in for the NinjaTrader instrument tick size appropriately. For stocks, the NinjaTrader default is 0.01. For forex instruments, the NinjaTrader default is 0.0001.

The product of these two user-supplied values — the *RenkoTickSize* and *instrument tick size* — sets the “range value” or “brick size” used in the renko bar creation computations.

Outwardly calm-looking, this spreadsheet is a lot like the duck on the pond — there is a lot happening under the surface. Yet it is all out of sight, driven by the VBA user-defined function code and user-defined class routines.

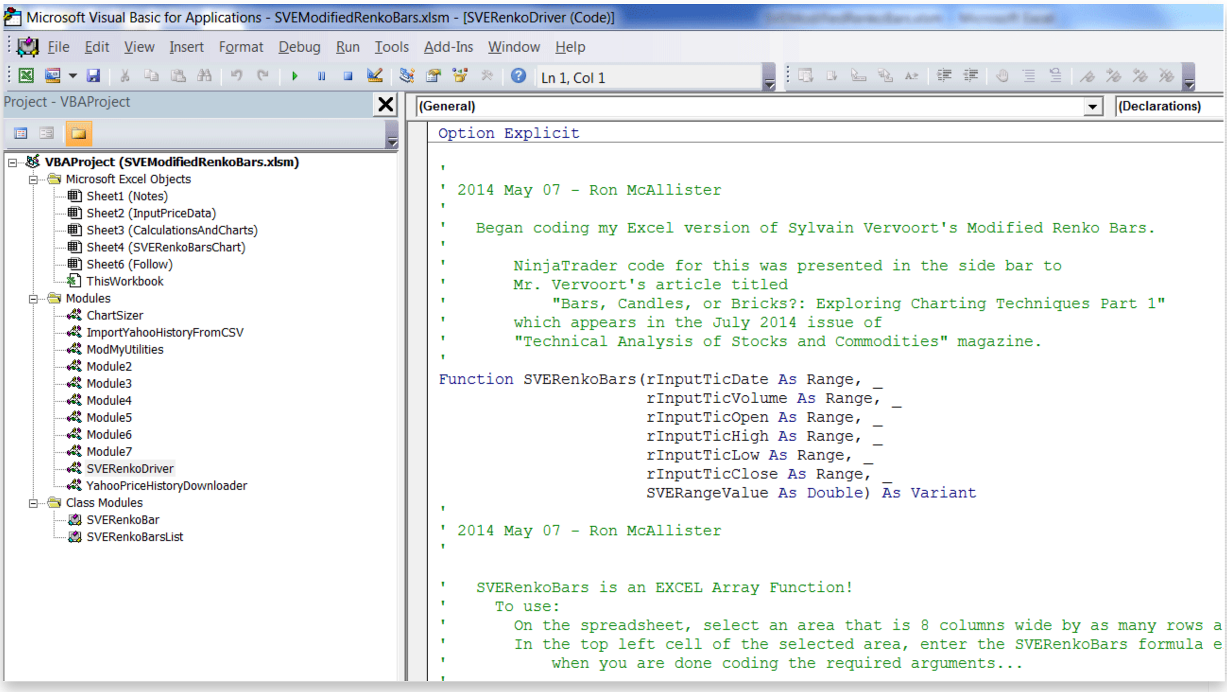
To get into the VBA editor and look at these routines, open the spreadsheet. Then press the ALT key and the F11 key at the same time. This should open the VBA editor (see Figure 10). If it does not, you may need to change a couple of your Excel macro-related options. For Windows-based Excel, the path to the macro-related settings is:

> File Tab→Options→Trust Center→Trust Center Settings→Macro Settings

On this panel, check the box next to “Trust access to the VBA object model.” And as long as you are already here, I recommend selecting “Disable all macros with notification” so that *you* get to decide if macros are allowed to run when you open a new spreadsheet for the first time. Close the *options* panels and return to the *home* tab.

Now ALT+F11 should open the VBA editor.

Next, if necessary, in the left column, click the “+” box next to the *modules* and *class modules* to expand the lists (Figure 10).


Microsoft Visual Basic for Applications - SVEModifiedRenkoBars.xlsm - [SVERenkoDriver (Code)]
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Ln 1, Col 1
Project - VBAProject
VBAProject (SVEModifiedRenkoBars.xlsm)
Microsoft Excel Objects
Sheet1 (Notes)
Sheet2 (InputPriceData)
Sheet3 (CalculationsAndCharts)
Sheet4 (SVERenkoBarsChart)
Sheet6 (Follow)
ThisWorkbook
Modules
ChartSizer
ImportYahooHistoryFromCSV
ModMyUtilities
Module2
Module3
Module4
Module5
Module6
Module7
SVERenkoDriver
YahooPriceHistoryDownloader
Class Modules
SVERenkoBar
SVERenkoBarsList
(General)
(Declarations)

```
Option Explicit

'
' 2014 May 07 - Ron McAllister
'
'   Began coding my Excel version of Sylvain Vervoort's Modified Renko Bars.
'
'       NinjaTrader code for this was presented in the side bar to
'       Mr. Vervoort's article titled
'           "Bars, Candles, or Bricks?: Exploring Charting Techniques Part 1"
'       which appears in the July 2014 issue of
'       "Technical Analysis of Stocks and Commodities" magazine.
'
Function SVERenkoBars(rInputTicDate As Range, _
                      rInputTicVolume As Range, _
                      rInputTicOpen As Range, _
                      rInputTicHigh As Range, _
                      rInputTicLow As Range, _
                      rInputTicClose As Range, _
                      SVERangeValue As Double) As Variant
'
' 2014 May 07 - Ron McAllister
'

'   SVERenkoBars is an EXCEL Array Function!
'     To use:
'       On the spreadsheet, select an area that is 8 columns wide by as many rows a
'       In the top left cell of the selected area, enter the SVERenkoBars formula e
'           when you are done coding the required arguments...
'
```

FIGURE 10: EXCEL, VBA EDITOR. You can browse the VBA code that was used to build the modified renko bars.

Doubleclick on SVERenkoDriver to open code for the user-defined array function mainline. Compare what you see in the SEVRenkoBars function to Vervoort’s NinjaTrader example code.

Doubleclick on SVERenkoBar to see the VBA-class definition for our renko bar object, and then on SVERenkoBarList to explore the VBA class methods used to create and manipulate our renko bar collection object, as the bars are built by the SEVRenkoBars function mainline and reported out to the spreadsheet at function completion.

A note on forex data: For what it’s worth, my “friend” Google and I have only found a few web-based sources for free tick-level forex data. And so far, my research implies that none of these will provide free one- and five-minute tick data older than 24 hours. Some will provide free *four*-hour ticks as old as three months. Many will provide end-of-day bars that go back several years. The *notes* tab of this month’s spreadsheet can direct you to one such website. You will need to perform a lot of manual copy & paste to get the data into a usable form for this spreadsheet or for any of my previous Traders’ Tips’ spreadsheets.

The spreadsheet file for this Traders’ Tip on modified renko bars can be downloaded here. To successfully download it, follow these steps:

- Right-click on the link to the Excel file SVEModifiedRenkoBars.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 July 2014 issue of
Technical Analysis of STOCKS & COMMODITIES magazine.

TRADERS' TIPS

August 2014

For this month's Traders' Tips, the focus is John Ehlers' article in this issue, "The Quotient Transform." Here we present the August 2014 Traders' Tips code with possible implementations in various software.

Code for TradeStation is already provided in Ehlers' article. 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 are contributed by various software developers or programmers for software that is capable of customization.

TRADESTATION: AUGUST 2014

In "The Quotient Transform" in this issue, author John Ehlers describes an early trend detection method that is designed to reduce the lag often found in other trend indicators. Ehlers already provides EasyLanguage code for TradeStation in his article for the early-onset trend-detection indicator and also describes an approach for creating a strategy based on this indicator.

For the convenience of TradeStation users, we're offering Ehlers' EasyLanguage code as well as an example strategy based on Ehlers' description given in his article as a downloadable file. To download the EasyLanguage code, please visit our TradeStation and EasyLanguage support forum. The code can be found at http://www.tradestation.com/TASC-2014, and is also shown below. The ELD filename for this code set is "_TASC_EarlyOnsetTrend.ELD."

_Ehlers_Early Onset Trend (Indicator)

```
inputs:
        LPPeriod( 30 ), K1( .85 ), K2( .4 ) ;
variables:
        alpha1( 0 ), HP( 0 ), a1( 0 ), b1( 0 ),
        c1( 0 ), c2( 0 ), c3( 0 ), Filt( 0 ), Peak(0),
        X( 0 ), Quotient1( 0 ), Quotient2( 0 ) ;

//Highpass filter cyclic components
//whose periods are shorter than
//100 bars
alpha1 = ( Cosine( .707 * 360 / 100 )
        + Sine ( .707 * 360 / 100 ) - 1 ) /
        Cosine( .707 * 360 / 100 ) ;


HP = ( 1 - alpha1 / 2 )*( 1 - alpha1 /
        2 )*( Close - 2 * Close[1] + Close[2] )
        + 2 * ( 1 - alpha1 ) * HP[1]
        - ( 1 - alpha1 ) * ( 1 - alpha1 ) * HP[2] ;

//SuperSmoother Filter
a1 = expvalue( -1.414 * 3.14159 / LPPeriod ) ;
b1 = 2 * a1 * Cosine( 1.414*180 / LPPeriod ) ;
c2 = b1 ;
c3 = -a1 * a1 ;
c1 = 1 - c2 - c3 ;
Filt = c1 * ( HP + HP[1] ) / 2 + c2 * Filt[1]
+ c3 * Filt[2] ;

//Fast Attack - Slow Decay Algorithm
Peak = .991 * Peak[1] ;
If AbsValue( Filt ) > Peak then
Peak = AbsValue( Filt ) ;

//Normalized Roofing Filter
if Peak <> 0 then
        X = Filt / Peak ;

Quotient1 = ( X + K1 ) / ( K1 * X + 1 ) ;
Quotient2 = ( X + K2 ) / ( K2 * X + 1 ) ;


Plot1( Quotient1, "Quotient1" ) ;
Plot2( Quotient2, "Quotient2" ) ;
Plot3( 0, "ZL" ) ;
```

_Ehlers_Early Onset Trend (Strategy)

```
inputs:
        LPPeriod( 30 ), K1( .85 ), K2( .4 ) ;
variables:
        alpha1( 0 ), HP( 0 ), a1( 0 ), b1( 0 ),
        c1( 0 ), c2( 0 ), c3( 0 ), Filt( 0 ), Peak(0),
        X( 0 ), Quotient1( 0 ), Quotient2( 0 ) ;

//Highpass filter cyclic components
//whose periods are shorter than
```

```
//100 bars
alpha1 = ( Cosine( .707 * 360 / 100 )
        + Sine ( .707 * 360 / 100 ) - 1 ) /
        Cosine( .707 * 360 / 100 ) ;


HP = ( 1 - alpha1 / 2 )*( 1 - alpha1 /
        2 )*( Close - 2 * Close[1] + Close[2] )
        + 2 * ( 1 - alpha1 ) * HP[1]
        - ( 1 - alpha1 ) * ( 1 - alpha1 ) * HP[2] ;

//SuperSmoother Filter
a1 = expvalue( -1.414 * 3.14159 / LPPeriod ) ;
b1 = 2 * a1 * Cosine( 1.414*180 / LPPeriod ) ;
c2 = b1 ;
c3 = -a1 * a1 ;
c1 = 1 - c2 - c3 ;
Filt = c1 * ( HP + HP[1] ) / 2 + c2 * Filt[1]
+ c3 * Filt[2] ;

//Fast Attack - Slow Decay Algorithm
Peak = .991 * Peak[1] ;
If AbsValue( Filt ) > Peak then
Peak = AbsValue( Filt ) ;

//Normalized Roofing Filter
if Peak <> 0 then
        X = Filt / Peak ;

Quotient1 = ( X + K1 ) / ( K1 * X + 1 ) ;
Quotient2 = ( X + K2 ) / ( K2 * X + 1 ) ;

if Quotient1 crosses over 0 then
        Buy next bar at Market ;

if Quotient2 crosses under 0 then
        Sell next bar at Market ;
```

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

A sample chart implementation is shown in Figure 1.

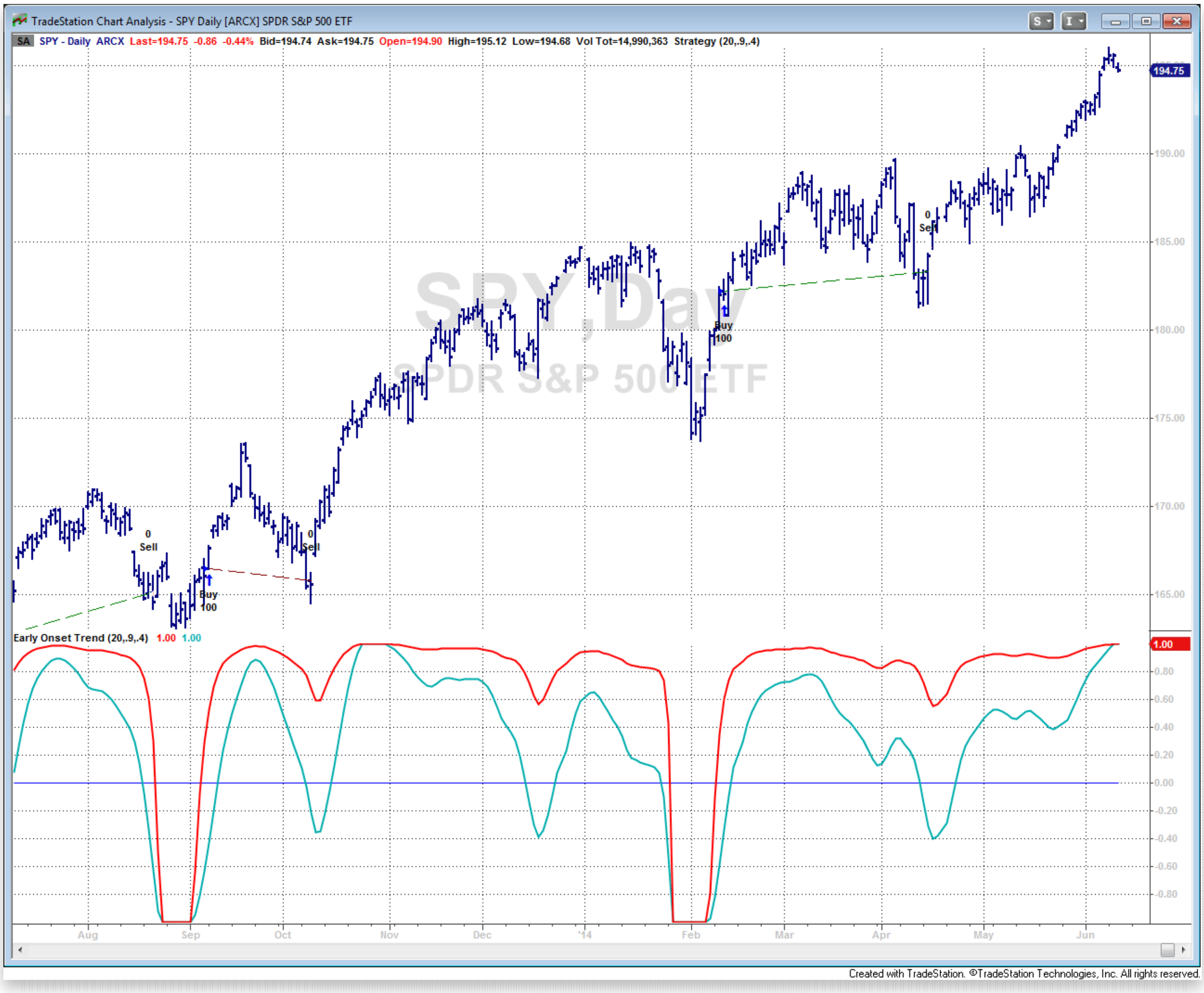


FIGURE 1: TRADESTATION. Here is a daily chart of the SPY showing the early onset trend indicator and strategy based on John Ehlers' article in this issue.

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.

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

BACK TO LIST

eSIGNAL: AUGUST 2014

For this month's Traders' Tip, we've provided the formula EarlyOnsetTrendIndicator.efs based on the formula described in John Ehlers' article in this issue, "The Quotient Transform."

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.

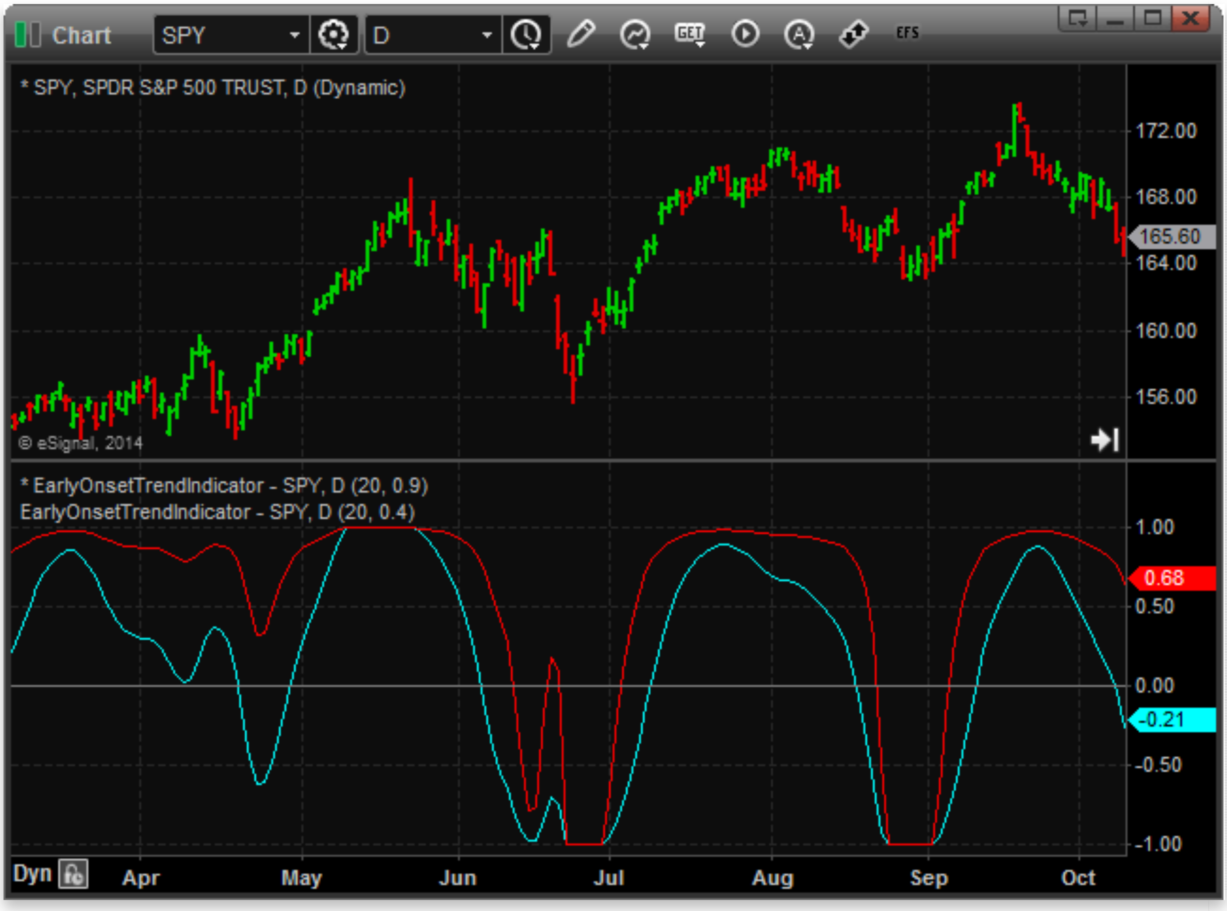


FIGURE 2: eSIGNAL. Here's an example of the EarlyOnsetTrendIndicator study applied to a chart of the SPY.

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 B   © 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:
    The Quotient Transform by John F. Ehlers
```

```
Formula Parameters:                     Default:
LPPeriod                                30
K                                       0.85

Version:            1.00  09/06/2014

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("EarlyOnsetTrendIndicator");
    setDefaultBarFgColor(Color.red);

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

    var x = 0;

    fpArray[x] = new FunctionParameter("fpLPPeriod", FunctionParameter.NUMBER);
    with(fpArray[x++])
    {
        setName("LPPeriod");
        setDefault(30);
        setLowerLimit(1);
    }

    fpArray[x] = new FunctionParameter("fpK", FunctionParameter.NUMBER);
    with(fpArray[x++])
    {
        setName("K");
        setDefault(0.85);
        setLowerLimit(-1);
        setUpperLimit(1);
    }
}

var bInit = false;
var bVersion = null;

xHP = null;
xFilt = null;
xPeak = null;
xX = null;
xQuotient = null;

function main(fpLPPeriod, fpK)
{
    if (bVersion == null) bVersion = verify();
    if (bVersion == false) return;

    if (!bInit)
    {
        xHP = efsInternal("Calc_HP");
        xFilt = efsInternal("Calc_Filt", xHP, fpLPPeriod);
        xPeak = efsInternal("Calc_Peak", xFilt);
        xX = efsInternal("Calc_X", xPeak, xFilt);
        xQuotient = getSeries(efsInternal("Calc_Quotient", xX, fpK));

        bInit = true;
    }

    return xQuotient;
}

var xClose = null;
var nAlpha1 = 0;

function Calc_HP()
{
    if (getBarState() == BARSTATE_ALLBARS)
    {
        xClose = close();

        nAlpha1 = (Math.cos((0.707 * 360 / 100) * (Math.PI / 180)) +
                   Math.sin((0.707 * 360 / 100) * (Math.PI / 180))  - 1) /
                   Math.cos((0.707 * 360 / 100) * (Math.PI / 180));

    }

    var nClose_0 = xClose.getValue(0);
    var nClose_1 = xClose.getValue(-1);
    var nClose_2 = xClose.getValue(-2);

    if (nClose_0 == null || nClose_1 == null || nClose_2 == null)
        return;

    var arrRefHP = ref(-1, -2);

    var nHP_1 = arrRefHP[0];
    var nHP_2 = arrRefHP[1];

    var nReturnValue = (1 - nAlpha1 / 2) * (1 - nAlpha1 / 2) * (nClose_0 - 2 *
nClose_1 + nClose_2) +
                       2 * (1 - nAlpha1) * nHP_1 -
                       (1 - nAlpha1) * (1 - nAlpha1) * nHP_2;

    return nReturnValue;
}

var nA1 = 0;
var nB1 = 0;
var nC1 = 0;
var nC2 = 0;
var nC3 = 0;

function Calc_Filt(xHP, nLPPeriod)
{
    if (getBarState() == BARSTATE_ALLBARS)
    {
        nA1 = Math.exp(-1.414 * 3.14159 / nLPPeriod);
        nB1 = 2 * nA1 * Math.cos((1.414 * 180 / nLPPeriod) * (Math.PI / 180));
        nC2 = nB1;
        nC3 = -nA1 * nA1;
        nC1 = 1 - nC2 - nC3;
    }
    var nHP_0 = xHP.getValue(0);
    var nHP_1 = xHP.getValue(-1);

    if (nHP_0 == null || nHP_1 == null)
        return;

    var arrRefFilt = ref(-1, -2);

    var nFilt_1 = arrRefFilt[0];
    var nFilt_2 = arrRefFilt[1];

    var nReturnValue = nC1 * (nHP_0 + nHP_1) / 2 + nC2 * nFilt_1 + nC3 * nFilt_2;

    return nReturnValue;
}

function Calc_Peak(xFilt)
{
    var nFilt = xFilt.getValue(0);

    if (nFilt == null)
        return;

    var nPeak_1 = ref(-1);

    var nPeak = 0.991 * nPeak_1;

    if (Math.abs(nFilt) > nPeak)
        nPeak = Math.abs(nFilt);

    var nReturnValue = nPeak;

    return nReturnValue;
}

function Calc_X(xPeak, xFilt)
{
    var nPeak = xPeak.getValue(0);
    var nFilt = xFilt.getValue(0);

    if (nPeak == null || nFilt == null)
        return;

    var nReturnValue = 0;

    if (nPeak != 0)
        nReturnValue = nFilt / nPeak;

    return nReturnValue;
}

function Calc_Quotient(xX, nK)
{
    var nX = xX.getValue(0);

    if (nX == null)
        return;

    var nReturnValue = (nX + nK) / (nK * nX + 1);
    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

THINKORSWIM: AUGUST 2014

In “The Quotient Transform” in this issue, author John Ehlers gives us a new tool for detecting trends and defining how long a trend will last. He introduces the idea of the *quotient transform*, which can be used with trend indicators for an early detection of trend. At thinkorswim, we have used our proprietary scripting language *thinkScript* to build a study and a strategy for detecting trends early using this method.

We have made the loading process extremely easy by providing links for them. For the *strategy*, simply go to http://tos.mx/xxFu8X and choose *backtest* in thinkorswim. For the *study*, go to http://tos.mx/146oNq and choose *save script to thinkorswim*, then choose to rename your study as “OnsetTrendDetector.” You can adjust the parameters of these within the edit *studies window* to fine-tune your variables.

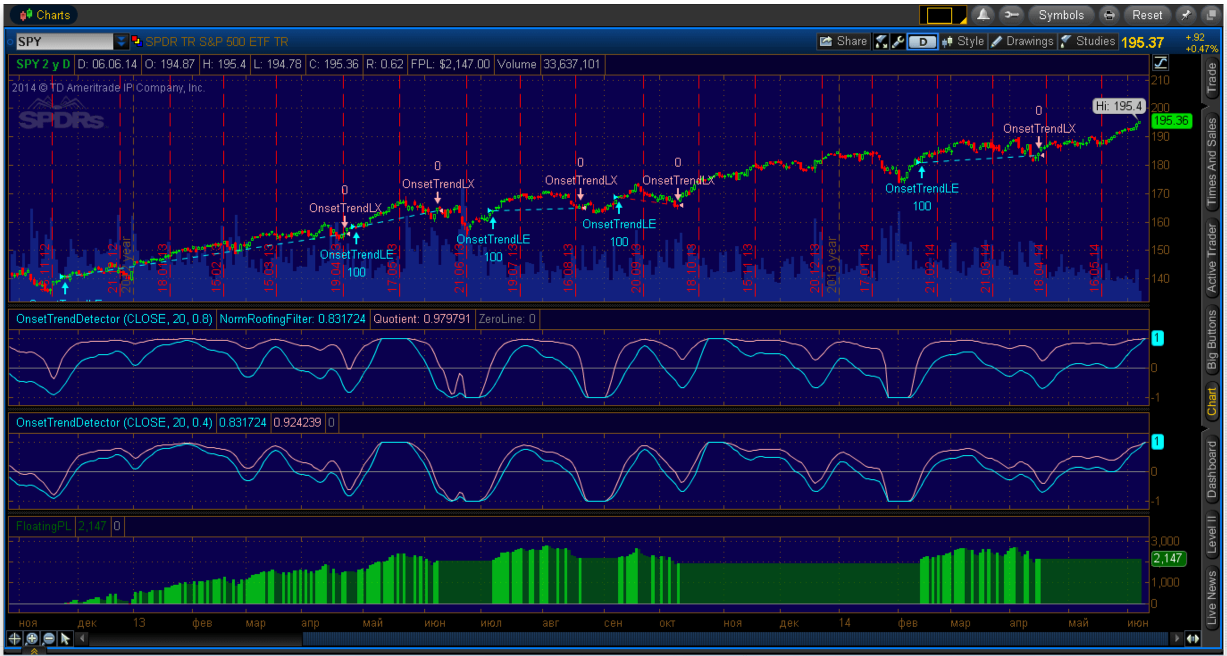


FIGURE 3: THINKORSWIM. Here are sample entry & exit points when the criteria described in Ehlers' article were met on a two-year daily chart of SPY.

The chart in Figure 3 shows entry & exit points when the criteria described in Ehlers' article were met for a two-year daily chart of SPY. The entry points displayed in blue on the price chart are defined by the top OnsetTrendDetector quotient crossing above zero. In the article, Ehlers suggests using a different *K* value for the exit, so the exit points are determined by the lower OnsetTrendDetector quotient crossing below zero.

This strategy can be tested with any product within thinkorswim to find your perfect opportunity. Happy swimming!

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

BACK TO LIST

WEALTH-LAB: AUGUST 2014

In his article in this issue, "The Quotient Transform," author John Ehlers introduces the quotient transform (QT), a zero-lag filter that can be used for the purpose of timely trend detection. The QT is an advancement of the technique he presented in his January 2014 S&C article, "Predictive And Successful Indicators." This time, the output of a roofing filter (which includes applying a high-pass filter and SuperSmoother filter) is normalized.

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

A drawback of the QT that Ehlers mentions in his article is that it tends to stay in trend mode for too long after the uptrend is over. But by applying two oscillators with different *K* parameters, Ehlers suggests that the system is facilitated to exit on or before the trend has run its course. To demonstrate the application of the new oscillator, we followed Ehlers' example and used two QTs. Our resulting trend-following system trades according to the rules as follows:

- Buy when the first 30-period QT with *K*=0.8 crosses zero from below
- Sell the position when the second 30-period QT with *K*=0.4 crosses zero from above.

See Figure 4 for an example of the trading system on a Wealth-Lab chart.

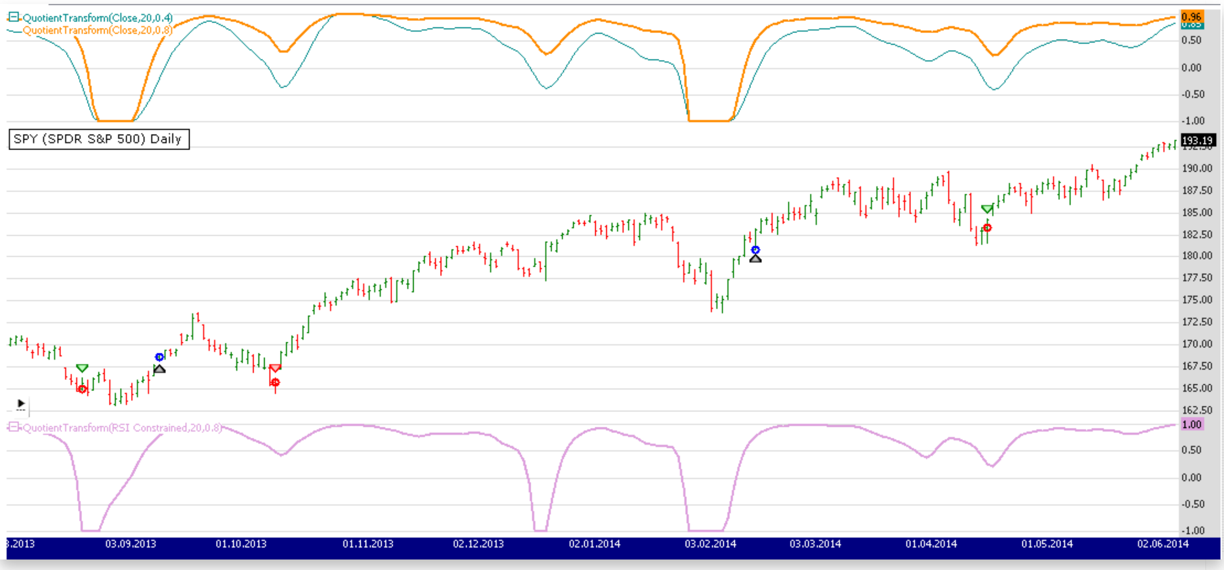


FIGURE 4: WEALTH-LAB, QT SYSTEM. This sample Wealth-Lab 6 chart illustrates application of the system's rules on a daily chart of SPY. The upper pane plots two QTs with *K* = 0.8 and 0.4. The bottom pane shows a constrained RSI processed with the quotient transform.

After applying the system to a portfolio of the 30 DJIA stocks (10 years of daily data, 10% equity per position, trading costs applied), we found that it had a drawdown too stressful and a market exposure too high for our taste, suggesting that a good portion of the accumulated profits are still being given back to the market. However, that doesn't lessen the fact that in general, the system was successful (although due to market's upside bias), beating buy & hold (115% vs. 103%) on 418 trades. Figure 5 shows an equity curve comparison.

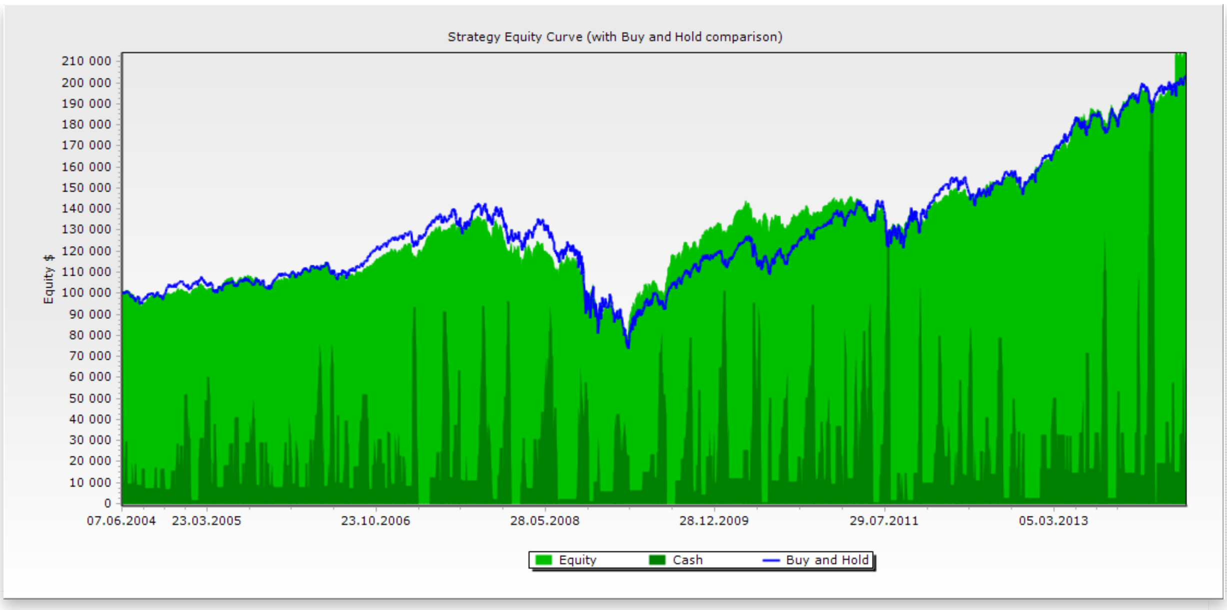


FIGURE 5: WEALTH-LAB, EQUITY CURVE COMPARISON OF SYSTEM VS. BUY & HOLD. The system equity was similar to buy & hold.

Figure 4 points at another shortcoming of this simple system: In October 2013, it exited a profitable trade that it had not reentered, missing a good trend ride. Thus, it seems that some sort of effective reentry technique wouldn't hurt the performance.

On a closing note, it would be intriguing to see if the *K* parameter could be made self-adjusting to the market's mood, as opposed to an arbitrary constant, which it is now.

C# Strategy 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 QuotientTransformTASC201408 : WealthScript
        {
                private StrategyParameter paramPeriod;
                private StrategyParameter paramK1;
                private StrategyParameter paramK2;

                public QuotientTransformTASC201408()
                {
                        paramPeriod = CreateParameter("Low-pass period",20,2,300,20);
                        paramK1 = CreateParameter("K for entries",0.8,-1,1,1);
                        paramK2 = CreateParameter("K for exits",0.4,-1,1,1);
                }
                protected override void Execute()
                {
                        QuotientTransform qtEn = 
QuotientTransform.Series(Close,paramPeriod.ValueInt,paramK1.Value);
                        QuotientTransform qtEx = 
QuotientTransform.Series(Close,paramPeriod.ValueInt,paramK2.Value);

                        for(int bar = 2; bar < Bars.Count; bar++)
                        {
                                if (IsLastPositionActive)
                                {
                                        if( CrossUnder( bar, qtEx, 0 ) )
                                                SellAtMarket( bar+1, LastPosition );
                                }
                                else
                                {
                                        if( CrossOver( bar, qtEn, 0 ) )
                                                if( BuyAtMarket( bar+1 ) != null )
                                                        LastPosition.Priority = 
-Close[bar];
                                }
                        }

                        HideVolume();
                        LineStyle ls = LineStyle.Solid;
                        ChartPane pQT = CreatePane(40,true,true);
                        PlotSeries(pQT,qtEx,Color.DarkCyan,ls,1);
                        PlotSeries(pQT,qtEn,Color.FromArgb(255,255,140,0),ls,2);

                        RSI rsi = RSI.Series(Close,14);
                        DataSeries qrsi = (rsi - 50) / 50;
                        ChartPane rp = CreatePane(40,false,true);
                        qrsi.Description = "RSI Constrained";
                        DataSeries qr = QuotientTransform.Series( qrsi, 
paramPeriod.ValueInt, paramK1.Value);
                        PlotSeries(rp,qr,Color.Plum,ls,2);
                }
        }
}
```

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

BACK TO LIST

CQG: AUGUST 2014

We’re providing CQG code for the early-onset trend indicator described in John Ehlers’ article in this issue, “The Quotient Transform.”

The early-onset trend indicator study has two parameters: *LPPeriod* and *K*, which may be configured in the “modify study parameters” window after the study has been applied to a chart in CQG. An example of the early-onset trend indicator for the SPY is depicted in the chart in Figure 6.

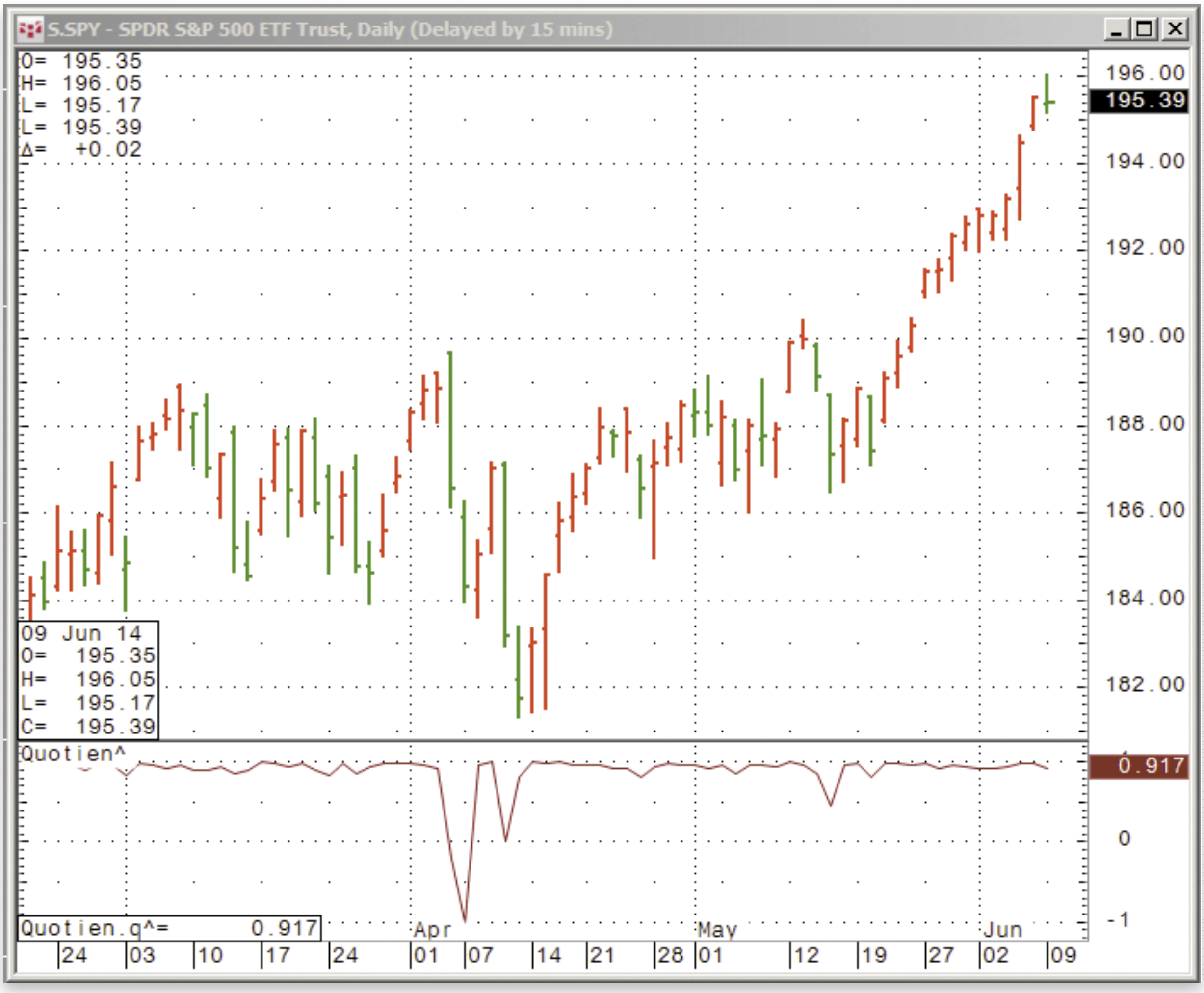


FIGURE 6: CQG. Here is an example of the study for the early-onset trend indicator.

CQG code for the study

```
/*Early-Onset Trend Indicator by John F. Ehlers*/
Parameters:
LPPeriod(15);
K(.95);
/*Highpass filter cyclic components whose periods are shorter than 100 bars*/
alpha1:= (Cos(.707*360/100) + Sin(.707*360/100) -1) / Cos(.707*360/100);
HP:=(1 - alpha1 /2)*(Close(@) - 2*Close(@)[-1] + Close(@)[-2])+ 2*(1 - alpha1)*HP[-1]
- (1 -alpha1)*(1 -
alpha1)*HP[-2];
/*SuperSmoother Filter*/
a1:= Exponential(-1.414*3.14159/LPPeriod);
b1:= 2* a1* Cos(1.414*180/LPPeriod);
c2:= b1;
c3:= -a1*a1;
c1:= 1 - c2 - c3;
Filt:= c1*(HP + HP[-1]) / 2 + c2*Filt[-1]+ c3*Filt[-2];
```

```
/*Fast Attack - Slow Decay Algorithm*/
Peak:= if(Abs(Filt) > Peak[-1], Abs(Filt),Peak[-1] * .991);
/*Normalized Roofing Filter*/
X:=Filt/Peak;
Quotient:= (X + K) / (K*X + 1);
quotient
```

To download the component PAC containing the complete formula code, visit CQG Workspaces. To discuss this study, users can visit the CQG Forums at http://www.cqgforums.com. Our team of expert product specialists can advise CQG users on the usage, application, and code for this study.

The PAC can also be downloaded here.

Trading and investment carry a high level of risk, and CQG, Inc. does not make any recommendations for buying or selling any financial instruments. We offer educational information on ways to use our CQG trading tools, but it is up to our customers and other readers to make their own trading and investment decisions or to consult with a registered investment advisor.

—CQG Inc., www.cqg.com

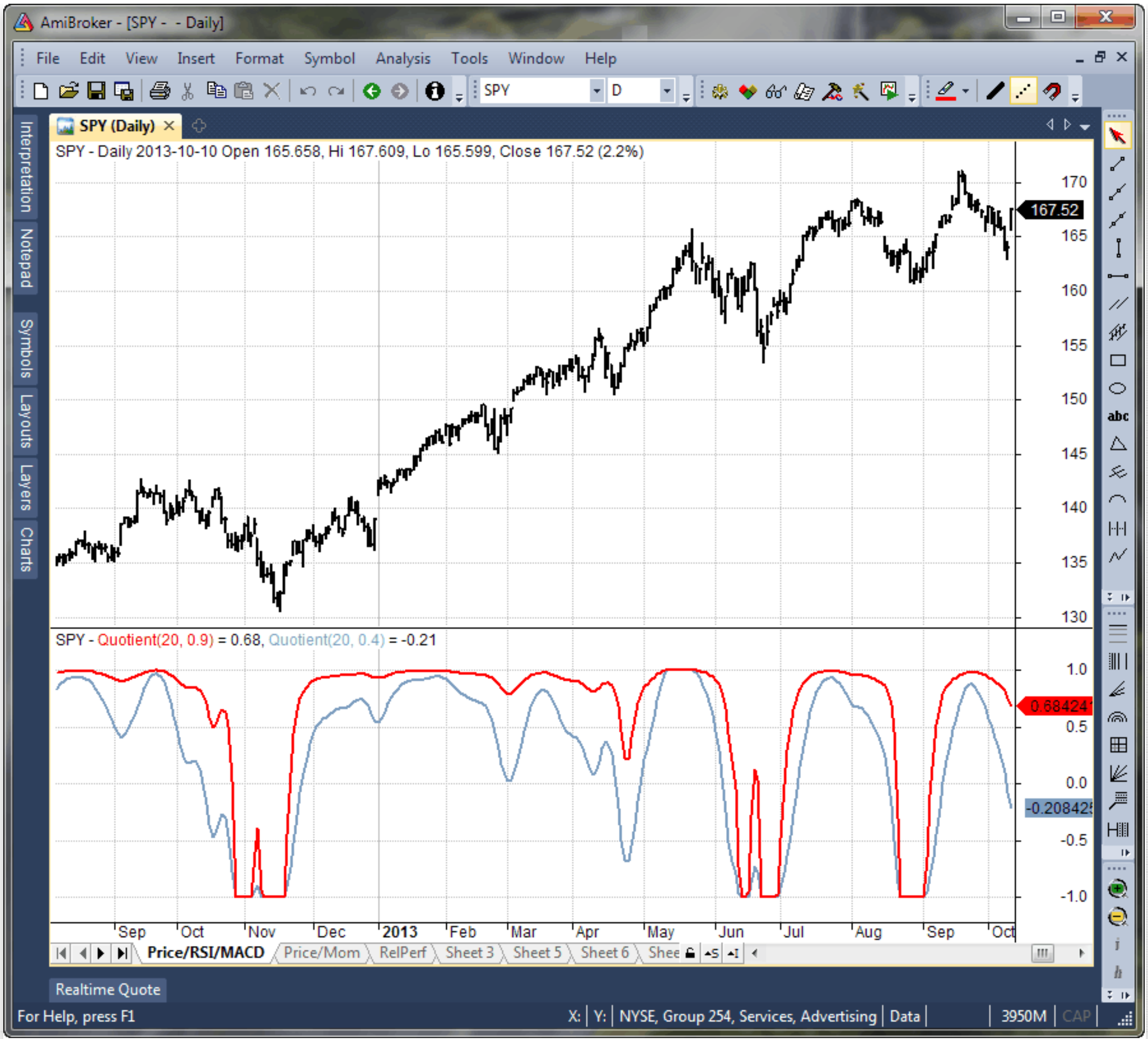
BACK TO LIST

AMIBROKER: AUGUST 2014

In “The Quotient Transform” in this issue, author John Ehlers presents a new indicator that is helpful for trend-following. We are providing a ready-to-use formula for AmiBroker (below). We have coded the quotient transform as a function so it can be called multiple times with different parameters.

```
SetBarsRequired( sbrAll );

function Quotient( LPPeriod, K )
{
    PI = 3.1415926;
    angle = 0.707 * 2 * PI / 100;
    alpha1 = ( cos( angle ) + sin( angle ) - 1 ) / cos( angle );

    a1 = exp( -1.414 * PI / LPPeriod );
    b1 = 2 * a1 * cos( 1.414 * PI / LPPeriod );
    c2 = b1;
    c3 = -a1 * a1;
    c1 = 1 - c2 - c3;

    HP = Close;
    Filt = HP;
    Pk = Filt;
    for( i = 2 ; i < BarCount; i++ )
    {
       HP[ i ] = ( ( 1 - alpha1 / 2 ) ^ 2 ) *
                ( Close[ i ] - 2 * Close[ i -1 ] + Close[ i - 2 ] ) +
                2 * ( 1 - alpha1 ) * HP[ i - 1 ] -
                HP[ i -2 ] * ( 1 - alpha1 ) ^ 2;

       Filt[ i ] = c1 * ( HP[ i ] + HP[ i - 1 ] )/2 +
                 c2 * Filt[ i - 1 ] +
                 c3 * Filt[ i - 2 ];

       Pk[ i ] = Max( 0.991 * Pk[ i - 1 ], abs( Filt[ i ] ) );
    }

    x = Nz( Filt / Pk );
    return ( X + K ) / ( K * X + 1 );
}

Plot( Quotient( 20, 0.9 ), "Quotient(20, 0.9)", colorRed, styleThick );
Plot( Quotient( 20, 0.4 ), "Quotient(20, 0.4)", colorLightBlue, styleThick );
```

A sample chart is shown in Figure 7.



AmiBroker - [SPY - - Daily]

SPY - Daily 2013-10-10 Open 165.658, Hi 167.609, Lo 165.599, Close 167.52 (2.2%)

SPY - Quotient(20, 0.9) = 0.68, Quotient(20, 0.4) = -0.21

For Help, press F1

FIGURE 7: AMIBROKER. A daily chart of SPY is in the upper pane while shown in the lower pane are two quotient transform indicators with two different parameter *K* values (0.9 and 0.4), based on John Ehlers' article in this issue.

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

BACK TO LIST

NEUROSHELL TRADER: AUGUST 2014

John Ehlers' quotient transform indicator described in his article in this issue can be easily implemented in NeuroShell Trader using NeuroShell Trader's ability to call external dynamic linked libraries (DLLs). Dynamic linked libraries may be written in C, C++, Power Basic, or Delphi.

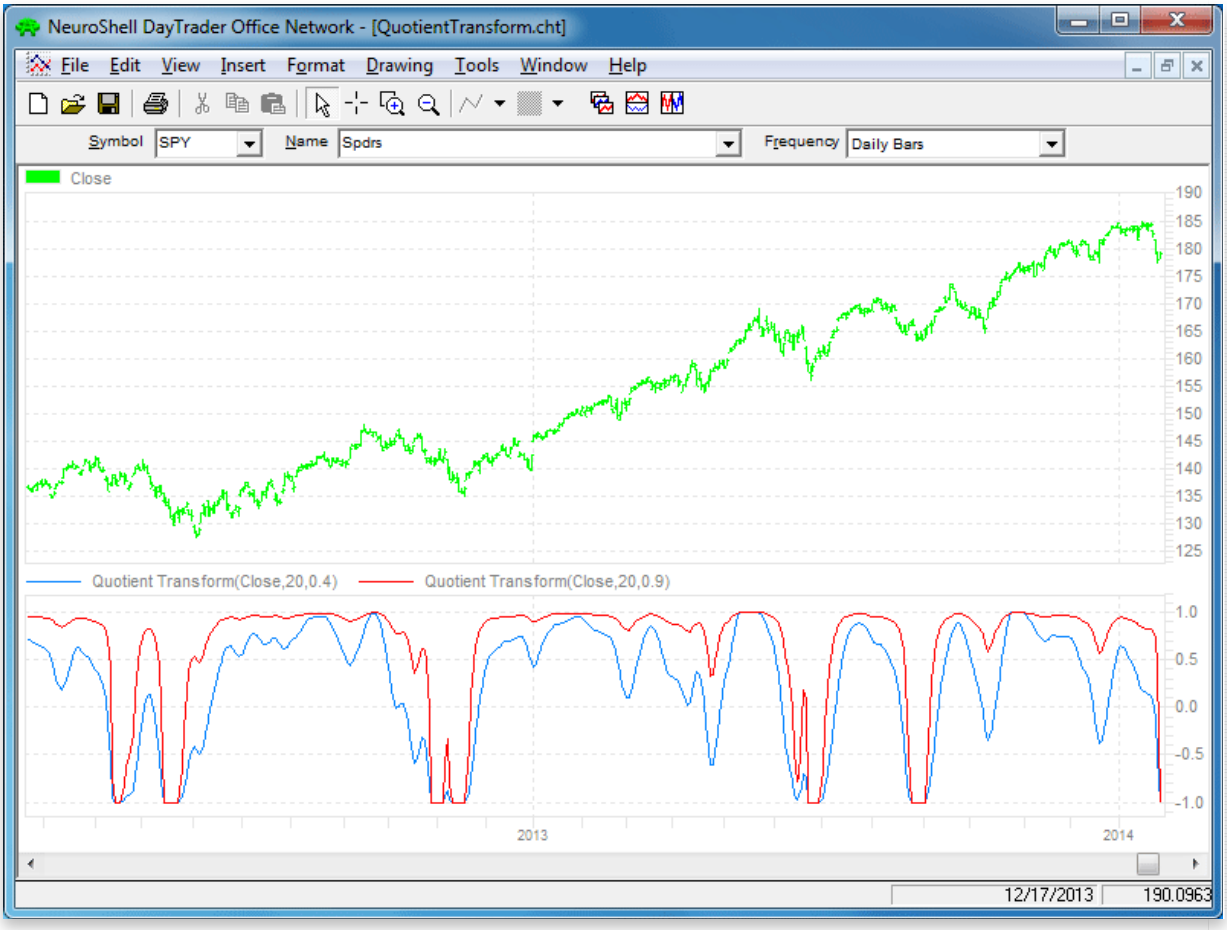
After moving the EasyLanguage code given in Ehlers' article to your preferred compiler and creating a DLL, you can insert the resulting indicators as follows:

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.

A dynamic trading system can be easily created in NeuroShell Trader by combining the quotient transform indicator with NeuroShell Trader's genetic optimizer to find optimal lengths. Similar filter- and cycle-based strategies can also be created using indicators found in 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 8.


NeuroShell DayTrader Office Network - [QuotientTransform.cht]

Symbol SPY

Name Spdrs

Frequency Daily Bars

12/17/2013

190.0963

FIGURE 8: NEUROSHELL TRADER. This NeuroShell Trader chart displays the quotient transform indicator.

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

BACK TO LIST

AIQ: AUGUST 2014

The AIQ code for this month is based on Dirk Vandycke's article in the March 2014 issue of STOCKS & COMMODITIES, "Expansions & Contractions, Part 1."

The code and EDS file can be downloaded from www.TradersEdgeSystems.com/traderstips.htm.

```
!EXPANSIONS & CONTRACTIONS
!Author: Dirk Vandycke, TASC March 2014
!Coded by: Richard Denning 6/11/2014
!www.TradersEdgeSystems.com

C is [close].
C1 is valresult(C,1).
H is [high].
L is [low].
O is [open].
TH is max(C1,H).
TL is min(C1,L).
TR is TH - TL.
ATR is simpleavg(TR,20).
RCL is (C-TL) / TR.
EU is (C-O) / TR.
```

Figure 9 shows an example of the expansive urge (EU) indicator, which was introduced by Vandycke in his article, on a chart of Rite-Aid (RAD).

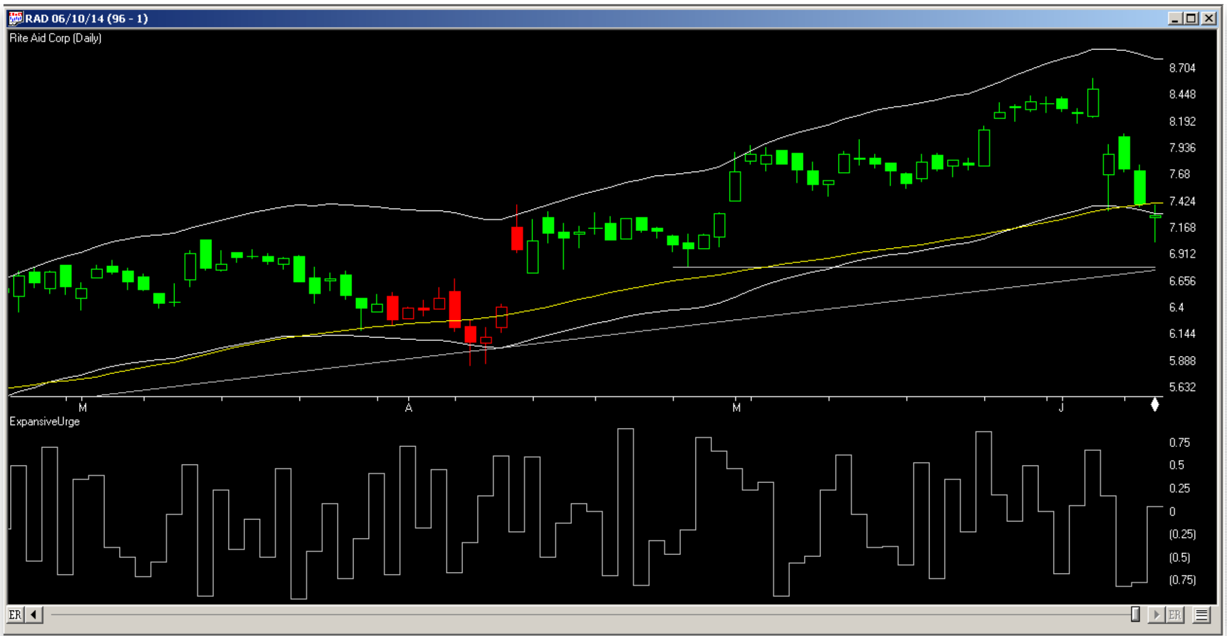


FIGURE 9: AIQ. Here is a sample application of Dirk Vandycke's expansive urge indicator (EU) on a chart of Rite-Aid (RAD).

—Richard Denning
info@TradersEdgeSystems.com
for AIQ Systems

BACK TO LIST

TRADERSSTUDIO: AUGUST 2014

The TradersStudio code I am providing this month is based on John Ehlers' article in this issue, "The Quotient Transform." The code is provided at the following websites:

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

The following files are provided in the download:

- **Function EHLERS_NORMROOF:** Returns the normalized roofing filter for cycles less than the *maxBarsLen* input
- **Function EHLERS_SUPSMO:** Returns a smoothed value based on an array series input and a smoothing length (*LPPeriod*)
- **Indicator plot:** Plots two smoothed, normalized roofing values based on the *Kslow* & *Kfast* inputs plus a zero line

- **System RSI_EXTENDED:** A trading system suggested by Ehlers using a fast and a slow smoothing of the indicator.

The system has the following rules:

- Buy the next bar at market open when the slow line crosses above zero.
- Exit the long position next bar at market open when the fast line crosses below zero.
- Reverse these rules for shorting (not coded here).

I ran the system on the S&P contract using data from Pinnacle Data Corp. Figure 10 shows the indicator with *Kslow* set to 0.95 and *Kfast* set to 0.80. Figure 11 shows the equity curve trading one contract of the S&P with the same parameter set for the period 4/21/1982 through 5/30/2014.

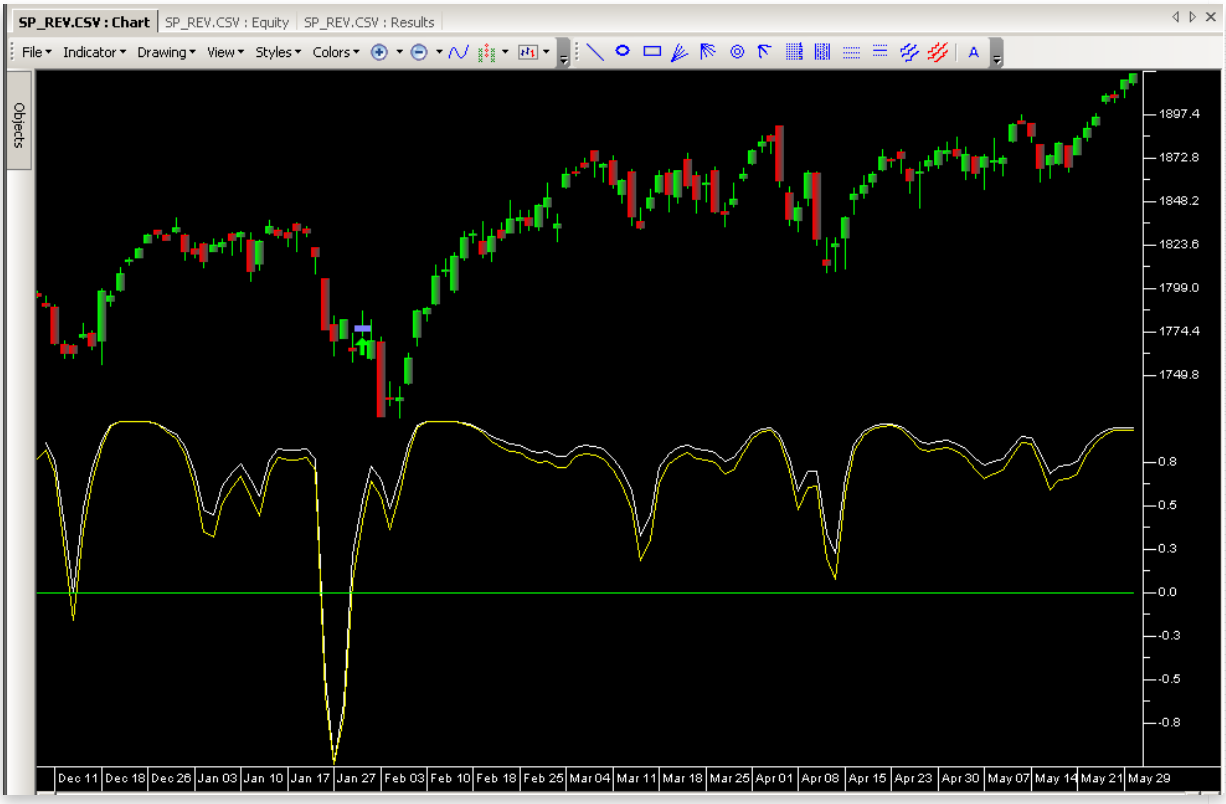


FIGURE 10: TRADERSSTUDIO, INDICATOR PLOT. This shows my "Indicator plot" downloadable code file plotting two smoothed, normalized roofing values on a chart of the S&P contract. The roofing values are based on the *Kslow* & *Kfast* inputs plus a zero line.

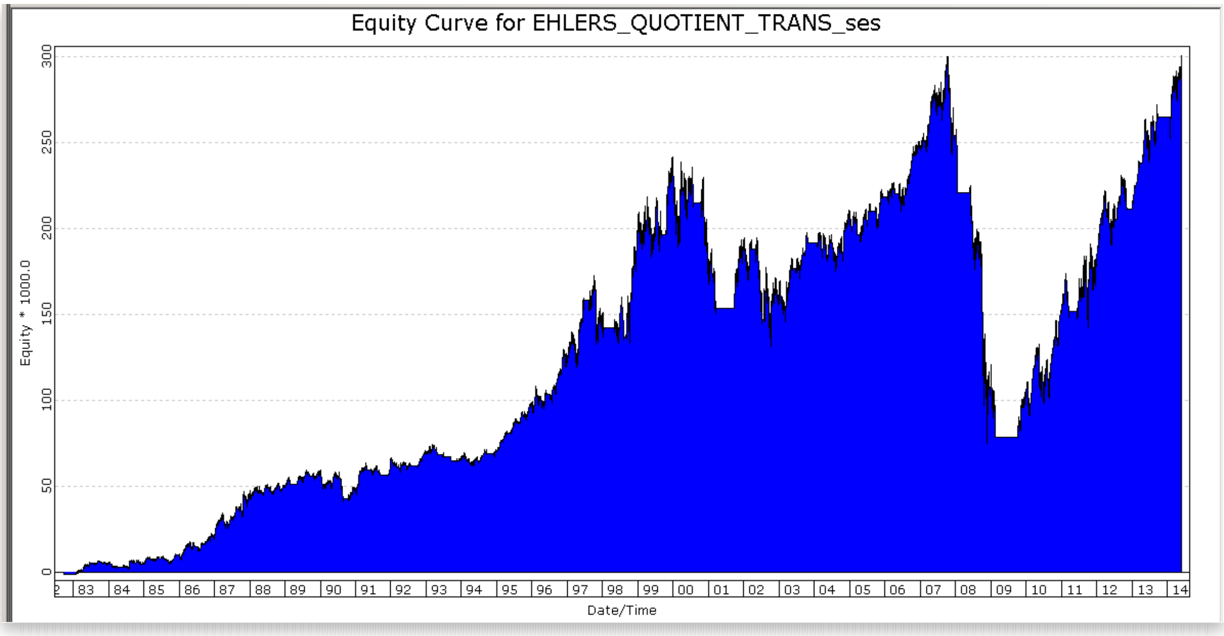


FIGURE 11: TRADERSSTUDIO, EQUITY CURVE. Here is a sample equity curv e trading one S&P contract per trade, long only, for the period April 1982 through May 2014.

```
'THE QUOTIENT TRANSFORM
'Author: John Ehlers, TASC August 2014
'Coded by: Richard Denning 6/08/2014
'www.TradersEdgeSystems.com

'Roofing filter for cycles less than maxBarsLen bars
Function EHLERS_NORMROOF(Price As BarArray, maxBarsLen, LPPeriod, K)
'Price=Close, maxBarsLen=100, LPPeriod=30, K=0.85
Dim alpha1
Dim HP As BarArray
Dim SSfilt As BarArray
Dim Peak As BarArray
Dim X As BarArray
Dim Quotient As BarArray
'Highpass filter cyclic components whose periods are shorter than 100 bars
  alpha1 = (Cos(DegToRad(0.707*360 / maxBarsLen) + Sin(DegToRad(0.707*360 /
maxBarsLen) - 1))) /Cos(DegToRad(0.707*360 / maxBarsLen))
  HP = (1 - alpha1 / 2)*(1 - alpha1 /2)*(Price - 2*Price[1] + Price[2]) + 2*(1 -
alpha1)*HP[1] - (1 - alpha1)*(1 - alpha1)*HP[2]
'Smooth with a Super Smoother Filter
  'assert(false)
  SSfilt = EHLERS_SUPSMO(HP,LPPeriod)
'Fast Attack - Slow Decay Algorithm
Peak = 0.991*Peak[1]
If Abs(SSfilt) > Peak Then Peak = Abs(SSfilt)
'Normalized Roofing Filter
If Peak<>0 Then X=SSfilt/Peak
if K*X+1 <> 0 then Quotient=(X+K)/(K*X+1)
EHLERS_NORMROOF = Quotient
End Function
'-------------------------------------------------------------------------
'SuperSmoother filter
```

```
' 2013 John F. Ehlers
Function EHLERS_SUPSMO(Price As BarArray,LPPeriod)
Dim a1, b1, c1, c2, c3, Filt As BarArray
a1 = Exp(-1.414*3.14159 / LPPeriod)
b1 = 2*a1*Cos(DegToRad(1.414*180 / LPPeriod))
c2 = b1
c3 = -a1*a1
c1 = 1 - c2 - c3
Filt = c1*(Price + Price[1]) / 2 + c2*Filt[1] + c3*Filt[2]
EHLERS_SUPSMO = Filt
End Function
'--------------------------------------------------------------------------
'Indicator plot for Early-Onset Trend (Norm Roofing Filter) Indicator
Sub EHLERS_NORMROOF_IND(Kslow,Kfast)
'Price=Close, maxBarsLen=100, LPPeriod=30, K=0.85
plot1(EHLERS_NORMROOF(Close,100,30,Kslow))
Plot2(EHLERS_NORMROOF(Close,100,30,Kfast))
plot3(0)
End Sub
'---------------------------------------------------------------------------
Sub EHLERS_QUOTIENT_TRANS_SYS(maxBarsLen,LPPeriod,Kslow,Kfast)
'Price=C,maxBarsLen=100,LPPeriod=30,Kslow=0.85,Kfast=0.40
Dim SlowLine As BarArray
Dim FastLine As BarArray
SlowLine = EHLERS_NORMROOF(Close,maxBarsLen,LPPeriod,Kslow)
FastLine = EHLERS_NORMROOF(Close,maxBarsLen,LPPeriod,Kfast)
If SlowLine > 0 And SlowLine[1] < 0 Then Buy("LE",1,0,Market,Day)
If FastLine < 0 And FastLine[1] > 0 Then ExitLong("LX","",1,0,Market,Day)
End Sub
'----------------------------------------------------------------------------
```

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

NINJATRADER: AUGUST 2014

The quotient transform indicator, which is introduced by John Ehlers in his article in this issue, “The Quotient Transform,” is available for download at www.ninjatrader.com/SC/August2014SC.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 indicator source code by selecting the menu Tools → Edit NinjaScript → Indicator from within the NinjaTrader Control Center window and selecting the “QuotientTransform” file.

A sample chart implementing the strategy is shown in Figure 12.

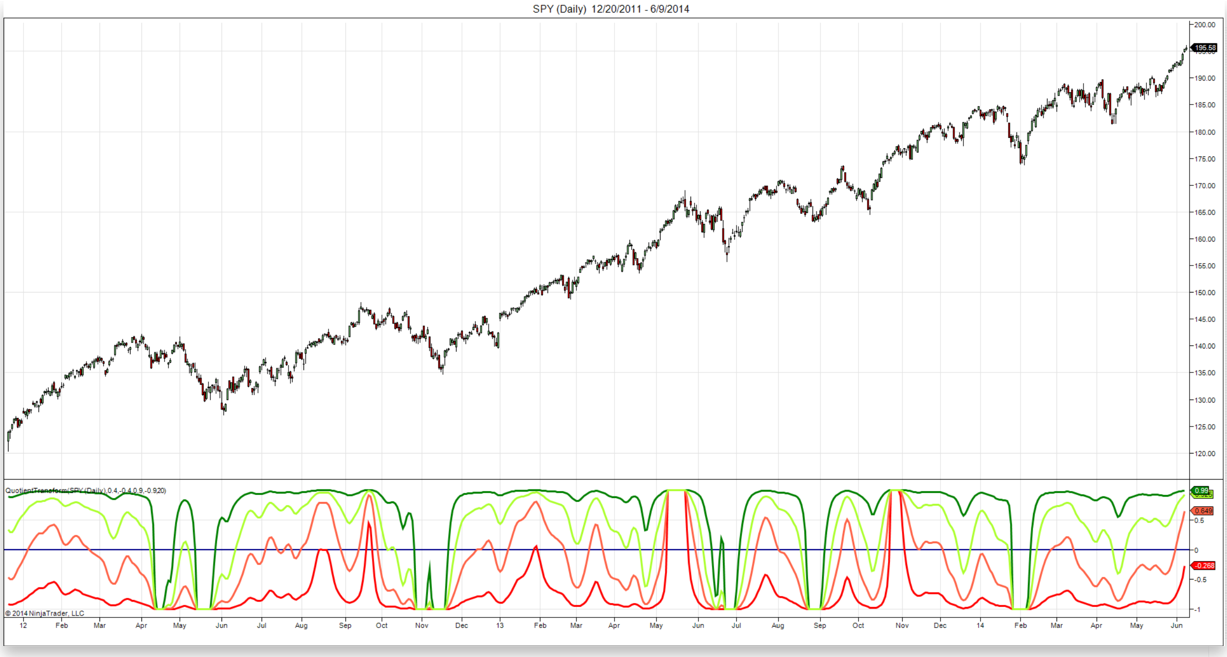


FIGURE 12: NINJATRADER. This NinjaTrader screenshot shows the quotient transform indicator applied to a daily SPY chart in NinjaTrader (note we are displaying both long and short bias plots as mentioned in Ehlers' article).

—Raymond Deux & Bertrand Wibbing
NinjaTrader, LLC, www.ninjatrader.com

BACK TO LIST

UPDATA: AUGUST 2014

Our Traders' Tip for this month is based on the article in this issue by John Ehlers, "The Quotient Transform."

In it, Ehlers develops an early trend detection indicator, utilizing a two-pole high-pass "roofing" filter that removes components of the price wave of longer wavelength and retaining only higher frequencies. The very high frequencies are removed via a SuperSmoother, which removes aliasing noise. The end result is a filter that provides a "roof" for allowed frequencies.

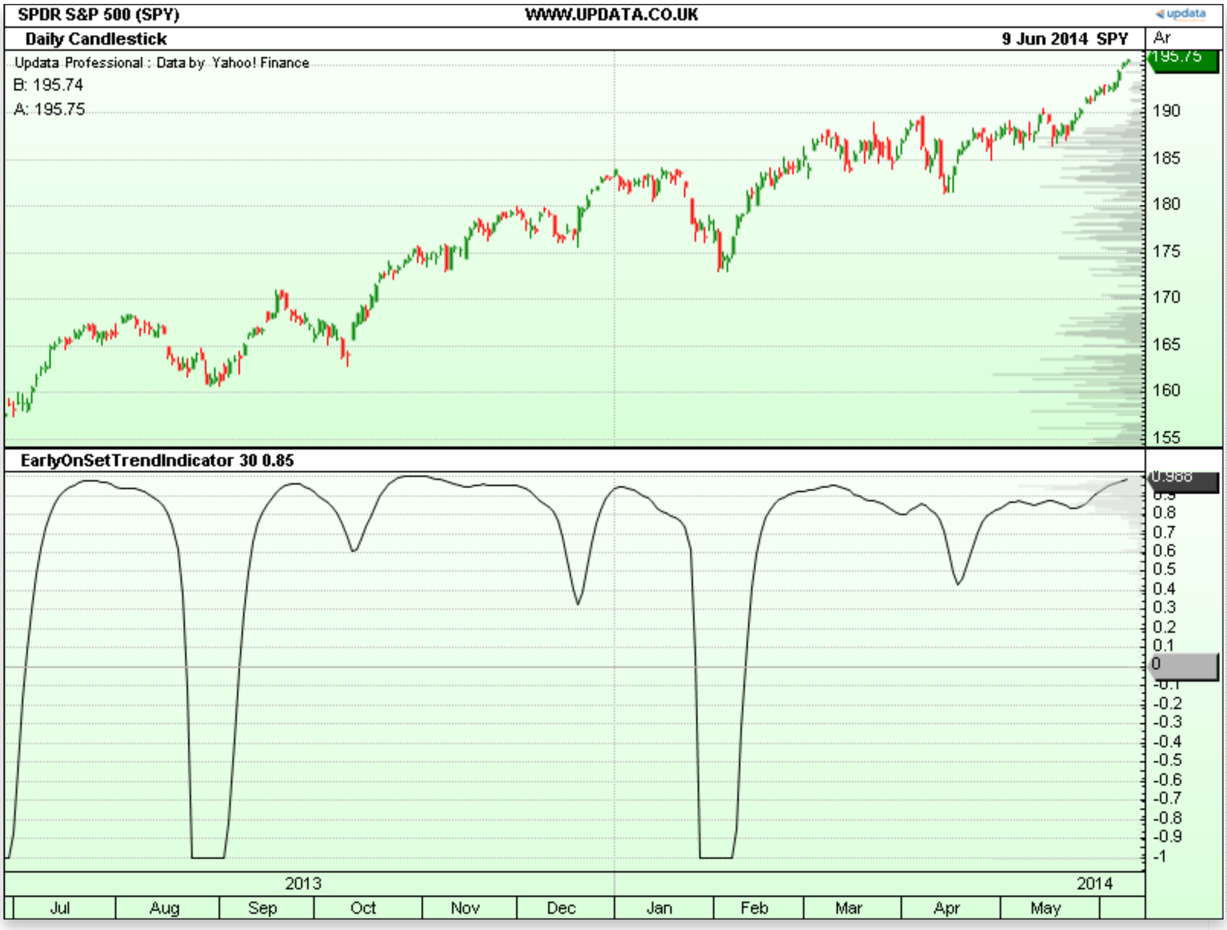


FIGURE 13: UPDATA. Here is an example of John Ehlers' early-onset trend indicator [30,0.85] as applied to the SPY ETF in daily resolution.

The Updata code based on Ehlers' article can be found in the Updata Library and may be downloaded by clicking the *custom* menu and Indicator Library. The code is also shown below for pasting into the Updata custom editor.

```
'EarlyOnSetTrendIndicator
PARAMETER "LP Period" #LPPeriod=30
PARAMETER "K" @K=0.85
DISPLAYSTYLE 2LINES
INDICATORTYPE CHART
NAME EarlyOnSetTrendIndicator
COLOUR2 RGB(180,180,180)
@ALPHA=0
@HP=0
@A1=0
@B1=0
@C1=0
@C2=0
@C3=0
@FILTER=0
@PEAK=0
@X=0
@QUOTIENT=0
```

```
@TWO_PI=0
@ONE_PI=0
FOR #CURDATE=#LPPeriod TO #CURDATE
    @TWO_PI=2*CONST_PI
    @ONE_PI=CONST_PI
    'HIGH PASS FILTER CYCLIC COMPONENTS SHORTER THAN 100 BARS
    @ALPHA=(COS(0.707*@TWO_PI/100)+SIN(0.707*@TWO_PI/100)-1)/COS(0.707*@TWO_PI/100)
    @HP=(1-@ALPHA/2)*(1-@ALPHA/2)*(CLOSE-2*CLOSE(1)+CLOSE(2))+2*
(1-@ALPHA)*HIST(@HP,1)-(1-@ALPHA)*(1-@ALPHA)*HIST(@HP,2)
    'SUPERSMOOTHER FILTER
    @A1=EXP(-1.414*@TWO_PI/#LPPeriod)
    @B1=2*@A1*COS(1.414*@ONE_PI/#LPPeriod)
    @C2=@B1
    @C3=-@A1*@A1
    @C1=1-@C2-@C3
    @FILTER=@C1*(@HP+HIST(@HP,1))/2+@C2*HIST(@FILTER,1)+@C3*HIST(@FILTER,2)
    'FAST ATTACK-SLOW DECAY ALGORITHM
    @PEAK=0.991*HIST(@PEAK,1)
    'NORMALISING ROOFING FILTER
    @PEAK=MAX(ABS(@FILTER),@PEAK)
    IF @PEAK !=0
       @X=@FILTER/@PEAK
    ENDIF
    @QUOTIENT=(@X+@K)/(@K*@X+1)
    @PLOT=@QUOTIENT
    @PLOT2=0

NEXT
```

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

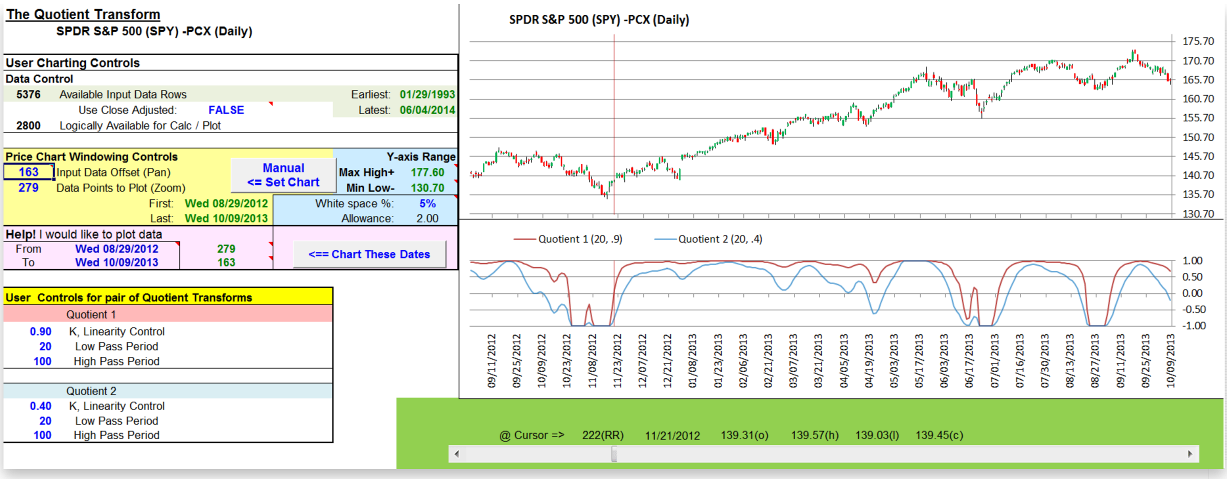
BACK TO LIST

MICROSOFT EXCEL: AUGUST 2014

In “The Quotient Transform” in this issue, author John Ehlers walks us through the development of a simple yet powerful tool for identifying the very early stages of an uptrend in an upwardly biased market.

For a simple entry/exit system, he suggests that a long entry should occur when *quotient 1* crosses above zero. He then adds a second copy of the indicator using a slightly less sensitive *linearity control* setting. Long position exits are signaled when *quotient 2* crosses below zero.

Using this exit strategy may take you out of a profitable trade before the major trend has run its course, as happens in Figure 14 when the second indicator drops below zero on 4/19/2013. A good trailing-stop strategy would have also taken you out.



The Quotient Transform
SPDR S&P 500 (SPY) -PCX (Daily)

User Charting Controls
Data Control
5376 Available Input Data Rows — Earliest: 01/29/1993
Use Close Adjusted: FALSE — Latest: 06/04/2014
2800 Logically Available for Calc / Plot

Price Chart Windowing Controls
163 Input Data Offset (Pan)
279 Data Points to Plot (Zoom)
First: Wed 08/29/2012
Last: Wed 10/09/2013
Y-axis Range
Max High+ 177.60
Min Low- 130.70
White space %: 5%
Allowance: 2.00

Help! I would like to plot data
From Wed 08/29/2012 279
To Wed 10/09/2013 163

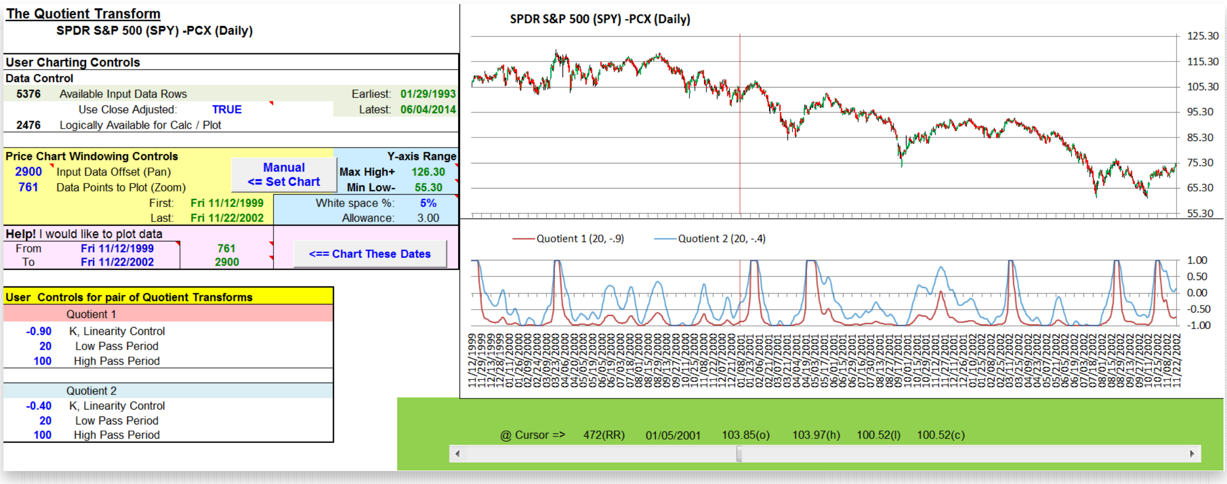
User Controls for pair of Quotient Transforms
Quotient 1
0.90 K, Linearity Control
20 Low Pass Period
100 High Pass Period
Quotient 2
0.40 K, Linearity Control
20 Low Pass Period
100 High Pass Period

@ Cursor => 222(RR) 11/21/2012 139.31(o) 139.57(h) 139.03(l) 139.45(c)

FIGURE 14: EXCEL, UPWARDLY BIASED MARKET. Here, the SPY is shown in an upwardly biased market, replicating Figure 3 from John Ehlers' article in this issue, "The Quotient Transform."

Ehlers points out that the indicator can also be used in a downward-biased market by using negative *linearity control* values.

Late 1999 to late 2002 was a period of downward market bias. In Figure 15, our signals are for short entry when *quotient 1* drops below zero. Close out the short position when *quotient 2* rises above zero.



The Quotient Transform
SPDR S&P 500 (SPY) -PCX (Daily)

User Charting Controls
Data Control
5376 Available Input Data Rows — Earliest: 01/29/1993
Use Close Adjusted: TRUE — Latest: 06/04/2014
2476 Logically Available for Calc / Plot

Price Chart Windowing Controls
2900 Input Data Offset (Pan)
761 Data Points to Plot (Zoom)
First: Fri 11/12/1999
Last: Fri 11/22/2002
Y-axis Range
Max High+ 126.30
Min Low- 55.30
White space %: 5%
Allowance: 3.00

Help! I would like to plot data
From Fri 11/12/1999 761
To Fri 11/22/2002 2900

User Controls for pair of Quotient Transforms
Quotient 1
-0.90 K, Linearity Control
20 Low Pass Period
100 High Pass Period
Quotient 2
-0.40 K, Linearity Control
20 Low Pass Period
100 High Pass Period

@ Cursor => 472(RR) 01/05/2001 103.85(o) 103.97(h) 100.52(l) 100.52(c)

FIGURE 15: EXCEL, DOWNWARD MARKET. With the SPY in a two-year period of downward bias, use negative *K* values and inverted signal logic.

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

TRADERS' TIPS

September 2014

For this month's Traders' Tips, the focus is Charlotte Hudgin's article in this issue, "Finding The Golden Triangle." Here we present the September 2014 Traders' Tips code with possible implementations in various software.

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

Here, you can read some discussion of the techniques' implementation by the Traders' Tips contributors as well as some example charts.

TRADESTATION: SEPTEMBER 2014

In "Finding The Golden Triangle" in this issue, author Charlotte Hudgin introduces her setup for identifying fast-growing stocks that have paused. She analyzes both price activity and volume while looking for a reversal at the bottom of a pullback.

Here, we are providing TradeStation EasyLanguage code for both an indicator and strategy based on Hudgin's ideas. The indicator can be used with a chart as well as with the TradeStation Scanner to search your symbol list of stocks. The author does not recommend a method for exiting her trades in the article, so when testing the strategy, you can apply one of the built-in TradeStation exit strategies or any other exit strategy that you choose. Please note that this code is compatible with TradeStation version 9.1 Update 20 and later.

Golden_Triangle (Indicator)

```
using elsystem ;
using elsystem.drawing ;
using elsystem.drawingobjects ;

inputs:
        Price( Close ),
        AVGLength( 50 ),
        PivotStrength( 3 ) ;

variables:
        MAValue( 0 ),
        PriceDiff( 0 ),
        PivotBar( 0 ),
        PivotPrice( 0 ),
        VolumeValue( 0 ),
        VolumeAvg( 0 ),
        PullBackBar( 0 ),
        WaitingForPivot( true ),
        WaitingForPullBack( false ),
        WaitingForConfirm( false ),
        SignalConfirmed( false ),
        InAChart( false ) ;

method void DrawPoint( int NumBarsAgo, color MyColor )
        variables: DTpoint MyPoint, TextLabel MyLabel ;
        begin
                begin
                MyPoint = DTPoint.Create( BarDateTime[NumBarsAgo],
                        High[NumBarsAgo] ) ;
                MyLabel = TextLabel.Create( MyPoint, "l" ) ;
                MyLabel.Color = MyColor ;
                MyLabel.Font = Font.Create( "Wingdings", 12 ) ;
                MyLabel.VStyle = VerticalStyle.Bottom ;
                MyLabel.HStyle = HorizontalStyle.Center ;
                MyLabel.Persist = false ;
                DrawingObjects.Add( MyLabel ) ;
                end ;
        end ;

method bool PivotFound()
        variables: bool Found ;
        begin
        if PivotHighVS( 1, High, PivotStrength,
                PivotStrength, PivotStrength + 1 ) <> -1 then
                Found = true
        else
                Found = false ;
        return Found ;
        end ;

method bool WhitespaceIncreasing()
        variables: bool WhiteSpaceOK ;
        begin
        if Close[20] > MAValue[20] and
                LinearRegValue( PriceDiff[PivotStrength],
                        20, PivotStrength )
                > LinearRegValue( PriceDiff[PivotStrength],
                        20, PivotStrength + 20 ) then
                WhiteSpaceOK = true
        else
                WhiteSpaceOK = false ;
        return WhiteSpaceOK ;
        end ;

method bool VolumeIncreasing( int NumBars )
        variables: bool VolumeInc, int Count ;
        begin
        count = 0 ;
        VolumeInc = false ;
        for Value1 = 0 to NumBars - 1
                begin
                if VolumeValue[Value1] > VolumeAvg[Value1] then
                        count += 1 ;
                end ;
        if Count >= NumBars * .5 then
                VolumeInc = true
        else
                VolumeInc = false ;
        return VolumeInc ;
        end ;

method int BarsSincePivot()
        begin
        return CurrentBar - PivotBar ;
        end ;

method int BarsSincePullBack()
        begin
        return CurrentBar - PullBackBar ;
        end ;

method bool VolumeConfirms ()
        begin
        return VolumeValue > VolumeAvg * 1.1 ;
        end ;

if BarType >= 2 and BarType < 5 then
        VolumeValue = Volume
else
        VolumeValue = Ticks ;

once
        begin
        InAChart = GetAppInfo( aiApplicationType ) = cChart ;
        end ;

MAValue = Average( Close, AVGLength ) ;
VolumeAvg = Average( VolumeValue, AVGLength ) ;
PriceDiff = ( Price - MAValue ) ;

if WaitingForPivot and PivotFound()
        and WhitespaceIncreasing() then
        begin
        if InAChart then
                DrawPoint( PivotStrength, Color.LightSeaGreen ) ;
        WaitingForPivot = false ;
        WaitingForPullBack = true ;
        PivotBar = BarNumber[PivotStrength] ;
        PivotPrice = High[PivotStrength] ;
        end ;

if WaitingForPullBack then
```

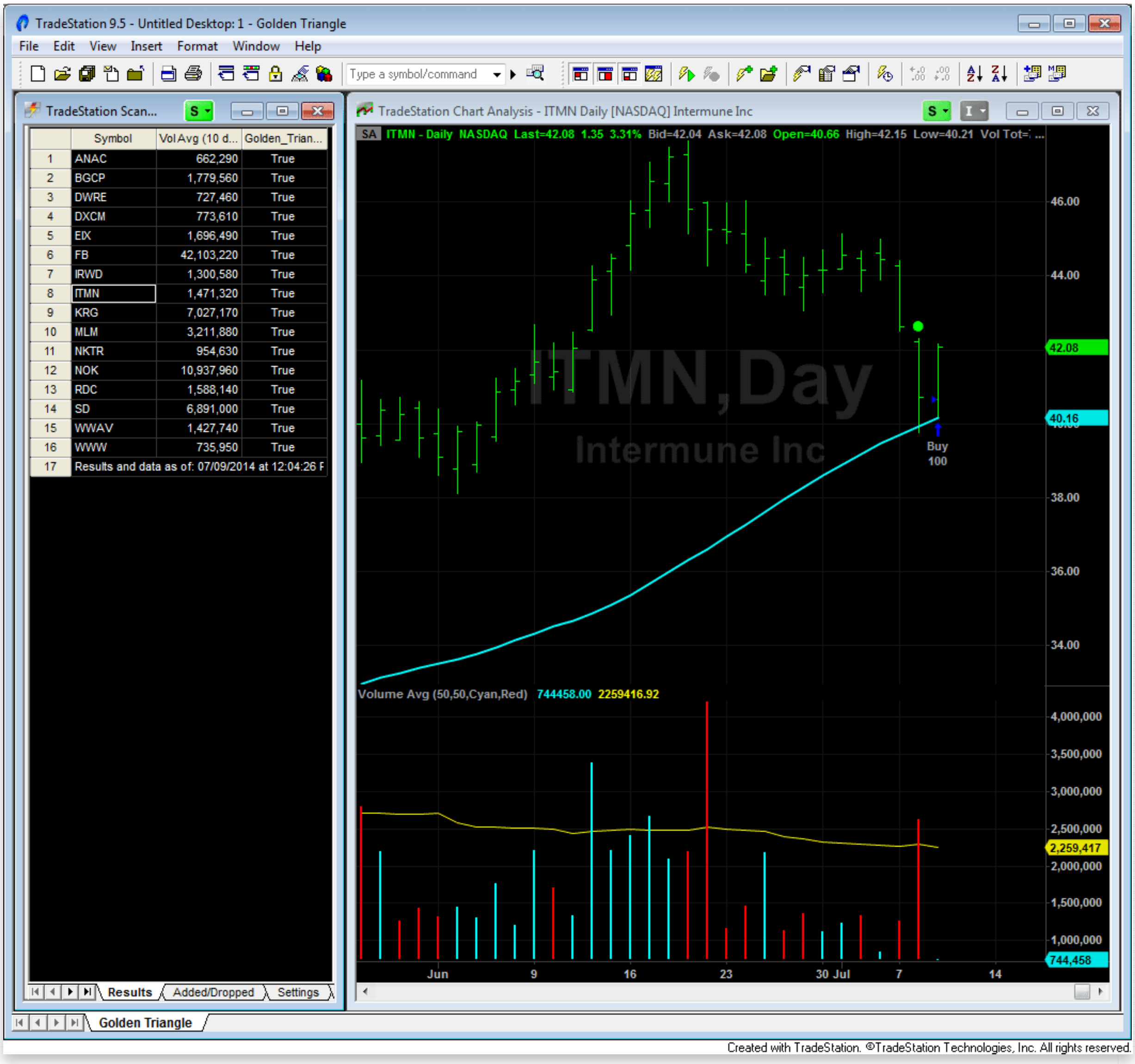
```
        begin
        if BarsSincePivot() > 20 then
                begin
                WaitingForPullBack = false ;
                WaitingForPivot = true ;
                end
        else if Low <= MAValue then
                begin
                PullBackBar = CurrentBar ;
                if BarsSincePivot() <= 3 then
                        begin
                        WaitingForPullBack = false ;
                        WaitingForConfirm = true ;
                        end
                else
                        begin
                        if VolumeIncreasing( BarsSincePivot() ) then
                                begin
                                WaitingForPullBack = false ;
                                WaitingForPivot = true ;
                                end
                        else
                                begin
                                WaitingForPullBack = false ;
                                WaitingForConfirm = true ;
                                end ;
                        end ;
                end ;
        end ;

if WaitingForConfirm then
        begin
        if BarsSincePullBack() > 15
                or Close > PivotPrice * 1.05 then
                begin
                WaitingForConfirm = false ;
                WaitingForPivot = true ;
                if InAChart then
                        DrawPoint( 0, Color.Coral ) ;
                end
        else
                begin
                if ( BarsSincePullBack() = 0 or Close > Close[1] )
                        and Close > MAValue and VolumeConfirms() then
                        begin
                        SignalConfirmed = true ;
                        WaitingForConfirm = false ;
                        if InAChart then
                                DrawPoint( 0, Color.Green ) ;
                        end ;
                end ;
        end ;

if SignalConfirmed then
        begin
        Alert( "Confirmed Signal" ) ;
        SignalConfirmed = false ;
        WaitingForPivot = true ;
        end ;
```

Golden_Triangle (Strategy)

```
using elsystem ;
using elsystem.drawing ;
using elsystem.drawingobjects ;

inputs:
        Price( Close ),
        AVGLength( 50 ),
        PivotStrength( 3 ) ;

variables:
        MAValue( 0 ),
        PriceDiff( 0 ),
        PivotBar( 0 ),
        PivotPrice( 0 ),
        VolumeValue( 0 ),
        VolumeAvg( 0 ),
        PullBackBar( 0 ),
        WaitingForPivot( true ),
        WaitingForPullBack( false ),
        WaitingForConfirm( false ),
        SignalConfirmed( false ),
        InAChart( false ) ;

method void DrawPoint( int NumBarsAgo, color MyColor )
        variables: DTpoint MyPoint, TextLabel MyLabel ;
        begin
                begin
                MyPoint = DTPoint.Create( BarDateTime[NumBarsAgo],
                        High[NumBarsAgo] ) ;
                MyLabel = TextLabel.Create( MyPoint, "l" ) ;
                MyLabel.Color = MyColor ;
                MyLabel.Font = Font.Create( "Wingdings", 12 ) ;
                MyLabel.VStyle = VerticalStyle.Bottom ;
                MyLabel.HStyle = HorizontalStyle.Center ;
                MyLabel.Persist = false ;
                DrawingObjects.Add( MyLabel ) ;
                end ;
        end ;

method bool PivotFound()
        variables: bool Found ;
        begin
        if PivotHighVS( 1, High, PivotStrength,
                PivotStrength, PivotStrength + 1 ) <> -1 then
                Found = true
        else
                Found = false ;
        return Found ;
        end ;

method bool WhitespaceIncreasing()
        variables: bool WhiteSpaceOK ;
        begin
        if Close[20] > MAValue[20] and
                LinearRegValue( PriceDiff[PivotStrength],
                        20, PivotStrength )
                > LinearRegValue( PriceDiff[PivotStrength],
                        20, PivotStrength + 20 ) then
                WhiteSpaceOK = true
        else
                WhiteSpaceOK = false ;
        return WhiteSpaceOK ;
        end ;

method bool VolumeIncreasing( int NumBars )
        variables: bool VolumeInc, int Count ;
        begin
        count = 0 ;
        VolumeInc = false ;
        for Value1 = 0 to NumBars - 1
                begin
                if VolumeValue[Value1] > VolumeAvg[Value1] then
                        count += 1 ;
                end ;
        if Count >= NumBars * .5 then
                VolumeInc = true
        else
                VolumeInc = false ;
        return VolumeInc ;
        end ;

method int BarsSincePivot()
        begin
        return CurrentBar - PivotBar ;
        end ;

method int BarsSincePullBack()
        begin
        return CurrentBar - PullBackBar ;
        end ;

method bool VolumeConfirms ()
        begin
        return VolumeValue > VolumeAvg * 1.1 ;
        end ;

if BarType >= 2 and BarType < 5 then
        VolumeValue = Volume
else
        VolumeValue = Ticks ;

once
        begin
        InAChart = GetAppInfo( aiApplicationType ) = cChart ;
        end ;

MAValue = Average( Close, AVGLength ) ;
VolumeAvg = Average( VolumeValue, AVGLength ) ;
PriceDiff = ( Price - MAValue ) ;

if WaitingForPivot and PivotFound()
        and WhitespaceIncreasing() then
        begin
        if InAChart then
                DrawPoint( PivotStrength, Color.LightSeaGreen ) ;
        WaitingForPivot = false ;
        WaitingForPullBack = true ;
        PivotBar = BarNumber[PivotStrength] ;
        PivotPrice = High[PivotStrength] ;
        end ;

if WaitingForPullBack then
        begin
        if BarsSincePivot() > 20 then
                begin
                WaitingForPullBack = false ;
                WaitingForPivot = true ;
                end
        else if Low <= MAValue then
                begin
                PullBackBar = CurrentBar ;
                if BarsSincePivot() <= 3 then
                        begin
                        WaitingForPullBack = false ;
                        WaitingForConfirm = true ;
                        end
                else
                        begin
                        if VolumeIncreasing( BarsSincePivot() ) then
                                begin
                                WaitingForPullBack = false ;
                                WaitingForPivot = true ;
                                end
                        else
                                begin
                                WaitingForPullBack = false ;
                                WaitingForConfirm = true ;
                                end ;
                        end ;
                end ;
        end ;

if WaitingForConfirm then
        begin
        if BarsSincePullBack() > 15
                or Close > PivotPrice * 1.05 then
                begin
                WaitingForConfirm = false ;
                WaitingForPivot = true ;
                if InAChart then
                        DrawPoint( 0, Color.Coral ) ;
                end
        else
                begin
                if ( BarsSincePullBack() = 0 or Close > Close[1] )
                        and Close > MAValue and VolumeConfirms() then
                        begin
                        SignalConfirmed = true ;
                        WaitingForConfirm = false ;
                        if InAChart then
                                DrawPoint( 0, Color.Green ) ;
                        end ;
                end ;
        end ;

if SignalConfirmed then
```

```
        begin
        Buy next bar at Market ;
        SignalConfirmed = false ;
        WaitingForPivot = true ;
        end ;
```

A sample chart is shown in Figure 1.



	Symbol	Vol Avg (10 d...	Golden_Trian...
1	ANAC	662,290	True
2	BGCP	1,779,560	True
3	DWRE	727,460	True
4	DXCM	773,610	True
5	EIX	1,696,490	True
6	FB	42,103,220	True
7	IRWD	1,300,580	True
8	ITMN	1,471,320	True
9	KRG	7,027,170	True
10	MLM	3,211,880	True
11	NKTR	954,630	True
12	NOK	10,937,960	True
13	RDC	1,588,140	True
14	SD	6,891,000	True
15	WWAV	1,427,740	True
16	WWW	735,950	True
17	Results and data as of: 07/09/2014 at 12:04:26 F		

Created with TradeStation. ©TradeStation Technologies, Inc. All rights reserved.

FIGURE 1: TRADESTATION. Here is a sample list of TradeStation Scanner results along with an example of the indicator and strategy applied to a daily chart of ITMN.

To download the EasyLanguage code, visit our TradeStation and EasyLanguage support forum. The code from this article can be found at http://www.tradestation.com/TASC-2014. The ELD filename is "_TASC_SEP2014_GOLDENTRIANGLE.ELD."

For more information about EasyLanguage in general, see http://www.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: SEPTEMBER 2014

For this month's Traders' Tip, we've provided the formula TheGoldenTriangleStrategy.efs based on the formula described in "Finding The Golden Triangle" by Charlotte Hudgin in this issue.

The study contains formula parameters that may be configured through the *edit chart* window (right-click on the chart and select "edit chart").

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, and can be downloaded here.

```
/*********************************
Provided By:
    Interactive Data Corporation (Copyright -í© 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:
    Finding The Golden Triangle by Charlotte Hudgin

Formula Parameters:                     Default:
Length SMA                              50
Consider Days of White Space            20
Consider Days to Compare Volume         0
Entry Position Color                    lime

Exit Position Color                     red

Version:            1.00  07/07/2014

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("TheGoldenTriangleStrategy");
    setPriceStudy(true);



    setCursorLabelName("Moving Average", 0);

    var x = 0;

    fpArray[x] = new FunctionParameter("fpLenSMA", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("Length SMA");
        setLowerLimit(1);
        setDefault(50);
    }

    fpArray[x] = new FunctionParameter("fpDaysWS", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("Consider Days of White Space");
        setLowerLimit(1);
        setDefault(20);
    }

    fpArray[x] = new FunctionParameter("fpDaysVolume", FunctionParameter.NUMBER);
    with(fpArray[x++]){
        setName("Consider Days to Compare Volume");
        setLowerLimit(0);
        setDefault(0);
    }



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


var xPriceSMA = null;
var xVolumeSMA = null;

var xHignesVol = null;



var nCurrDaysWS = 0;

var nLotSize = 0;



var bPriceConfirm = false;

var bVolumeConfirm = false;



var nPivotPrice = null;


function main(fpLenSMA, fpDaysWS, fpDaysVolume, fpEntryColor, fpExitColor){


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

    if (!bInit){



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

        xVolume = volume();



        if (fpDaysVolume != 0)

            xHignesVol = highest(fpDaysVolume, xVolume);

        xPriceSMA = sma(fpLenSMA);

        xVolumeSMA = sma(fpLenSMA, xVolume);



        nLotSize = Strategy.getDefaultLotSize();

        bInit = true;
    }
    if (getBarState() == BARSTATE_ALLBARS){

        bPriceConfirm = false;

        bVolumeConfirm = false;



        nCurrDaysWS = 0;

    };


    var nOpen = xOpen.getValue(0);
    var nHigh = xHigh.getValue(0);
    var nLow  = xLow.getValue(0);
    var nClose = xClose.getValue(0);

    var nPriorClose = xClose.getValue(-1);

    var nVolume = xVolume.getValue(0);



    var nPriceSMA = xPriceSMA.getValue(0);

    var nVolumeSMA = xVolumeSMA.getValue(0);



    var nHignesVol = 0;



    if (fpDaysVolume != 0)

        nHignesVol = xHignesVol.getValue(-1);


    if (nPriorClose == null || nPriceSMA == null || nVolumeSMA == null || nHignesVol
== null)
        return;



    if (getCurrentBarIndex() != 0){



        if (Strategy.isInTrade() && nHigh >= nPivotPrice){



            var nExitPrice = Math.max(nOpen, nPivotPrice);



            Strategy.doSell("Exit", Strategy.LIMIT, Strategy.THISBAR,
Strategy.DEFAULT,
nExitPrice);

            drawShapeRelative(0, AboveBar1, Shape.DOWNTRIANGLE, null, fpExitColor,
Text.PRESET, getCurrentBarIndex() + "Exit");
            drawTextRelative(0, AboveBar2, "Exit", fpExitColor, null,
Text.PRESET|Text.CENTER, null, null, getCurrentBarIndex() + "Exit");
            drawTextRelative(0, AboveBar3, nLotSize + " @ " +
formatPriceNumber(nExitPrice), fpExitColor, null, Text.PRESET|Text.CENTER, null,
null, getCurrentBarIndex() + "ExitSettings");



            bPriceConfirm = false;

            bVolumeConfirm = false;

        };



        if (!bPriceConfirm && nCurrDaysWS >= fpDaysWS && nLow <= nPriceSMA && nClose
>= nPriceSMA){



            bPriceConfirm = true;



            for (var j = 0; j > -getCurrentBarCount()-1; j--){



                nPivotPrice = xHigh.getValue(j-1);



                if (xHigh.getValue(j) > xHigh.getValue(j-1)){

                    nPivotPrice = xHigh.getValue(j);

                    break;

                }

            }



            if (nVolume > nVolumeSMA && nVolume > nHignesVol)

                bVolumeConfirm = true;

        };
        if (bPriceConfirm && !bVolumeConfirm){

            if (nVolume > nVolumeSMA && nVolume > nHignesVol && nClose > nPriceSMA &&
nClose > nPriorClose)

                 bVolumeConfirm = true;

        };



        if (bPriceConfirm && bVolumeConfirm && !Strategy.isInTrade()){



            var nEntryPrice = xOpen.getValue(1);



            if (nEntryPrice > nPivotPrice){

                bPriceConfirm = false;

                bVolumeConfirm = false;

            } else{

                Strategy.doLong("Entry", Strategy.MARKET, Strategy.NEXTBAR,
Strategy.DEFAULT);

                drawShapeRelative(1, BelowBar1, Shape.UPTRIANGLE, null, fpEntryColor,
Text.PRESET, getCurrentBarIndex() + "Entry");
                drawTextRelative(1, BelowBar2, "Entry", fpEntryColor, null,
Text.PRESET|Text.CENTER, null, null, getCurrentBarIndex() + "Entry");
                drawTextRelative(1, BelowBar3, nLotSize + " @ " +
formatPriceNumber(nEntryPrice), fpEntryColor, null, Text.PRESET|Text.CENTER, null,
null, getCurrentBarIndex() + "EntrySettings");

            };

        };

    };



    if (nLow > nPriceSMA)

        nCurrDaysWS ++

    else

        nCurrDaysWS = 0;




    return nPriceSMA;
}
```

```
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;
}
```

A sample chart is shown in Figure 2.

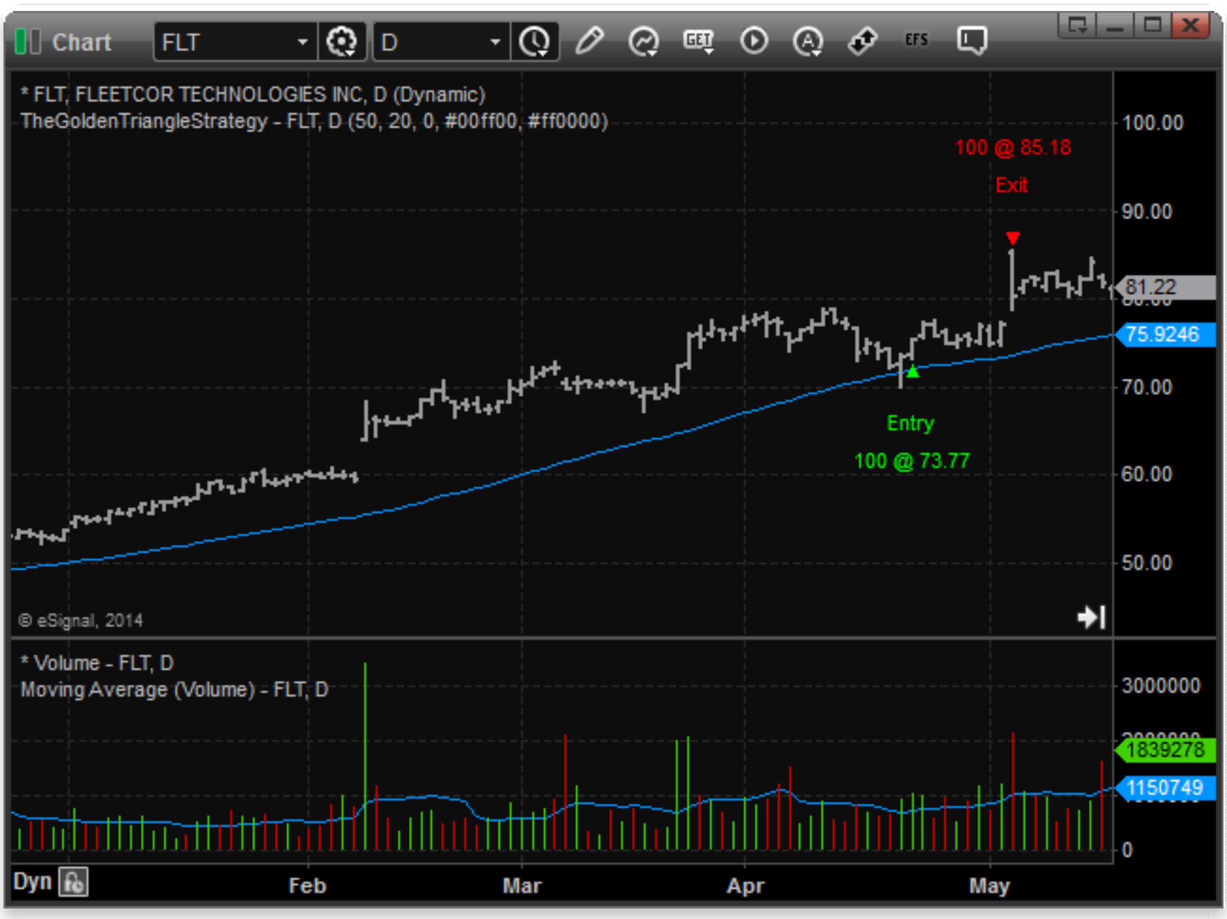


FIGURE 2: eSIGNAL. Here's an example of the

strategy on FleetCor Technologies Inc. (FLT).

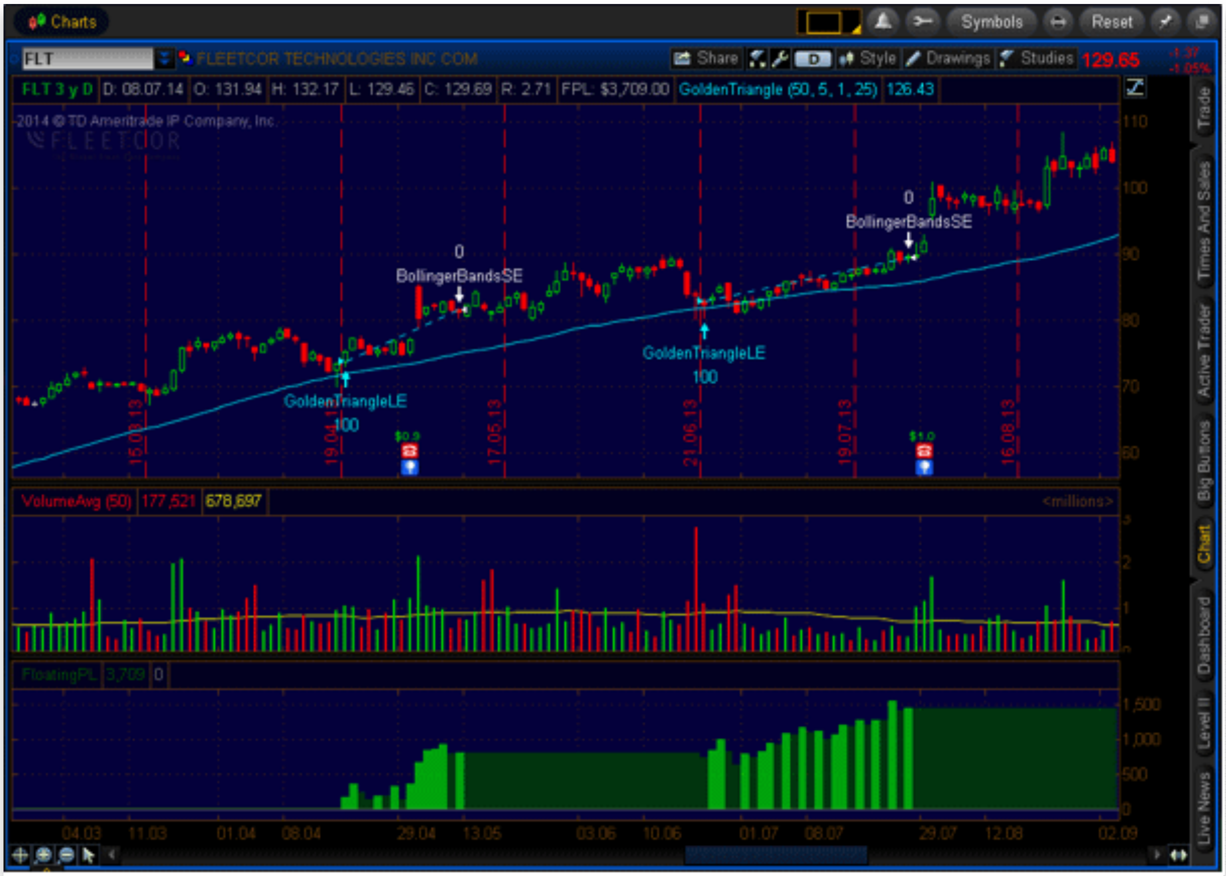
—Eric Lippert
eSignal, an Interactive Data company
800 779-6555, www.eSignal.com

BACK TO LIST

THINKORSWIM: SEPTEMBER 2014

In “Finding The Golden Triangle” in this issue, author Charlotte Hudgin defines a new strategy that indicates when a stock is likely to remain in a trend. At thinkorswim, we have used our proprietary scripting language thinkScript to build a strategy for detecting trends based on this method.

For simplicity in implementing it, we are offering the custom study downloadable from http://tos.mx/l7ox0B. Choose *backtest in thinkorswim*, then rename the study “GoldenTriangle.” You can adjust the parameters of these within the *edit studies* window to fine-tune your variables.



FLT 3 y D | D: 08.07.14 | O: 131.94 | H: 132.17 | L: 129.46 | C: 129.69 | R: 2.71 | FPL: $3,709.00 | GoldenTriangle (50, 5, 1, 25) | 126.43

VolumeAvg (50) | 177,521 | 678,697

FloatingPL | 3,709 | 0

FIGURE 3: THINKORSWIM. This three-year daily chart of FleetCor Technologies Inc. (FLT) shows sample entry points for Hudgin's described strategy. An existing Bollinger Band-based strategy is used for the exit point for demonstration purposes. The green histogram shows the strategy's performance over time.

The chart in Figure 3 shows entry points for the strategy described in Hudgin's article displayed on a three-year daily chart of FleetCor Technologies Inc. (FLT). You can see that this thinkorswim chart uses an existing Bollinger Band-based strategy for the exit point. The green histogram shows the strategy's performance over time.

Happy swimming!

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

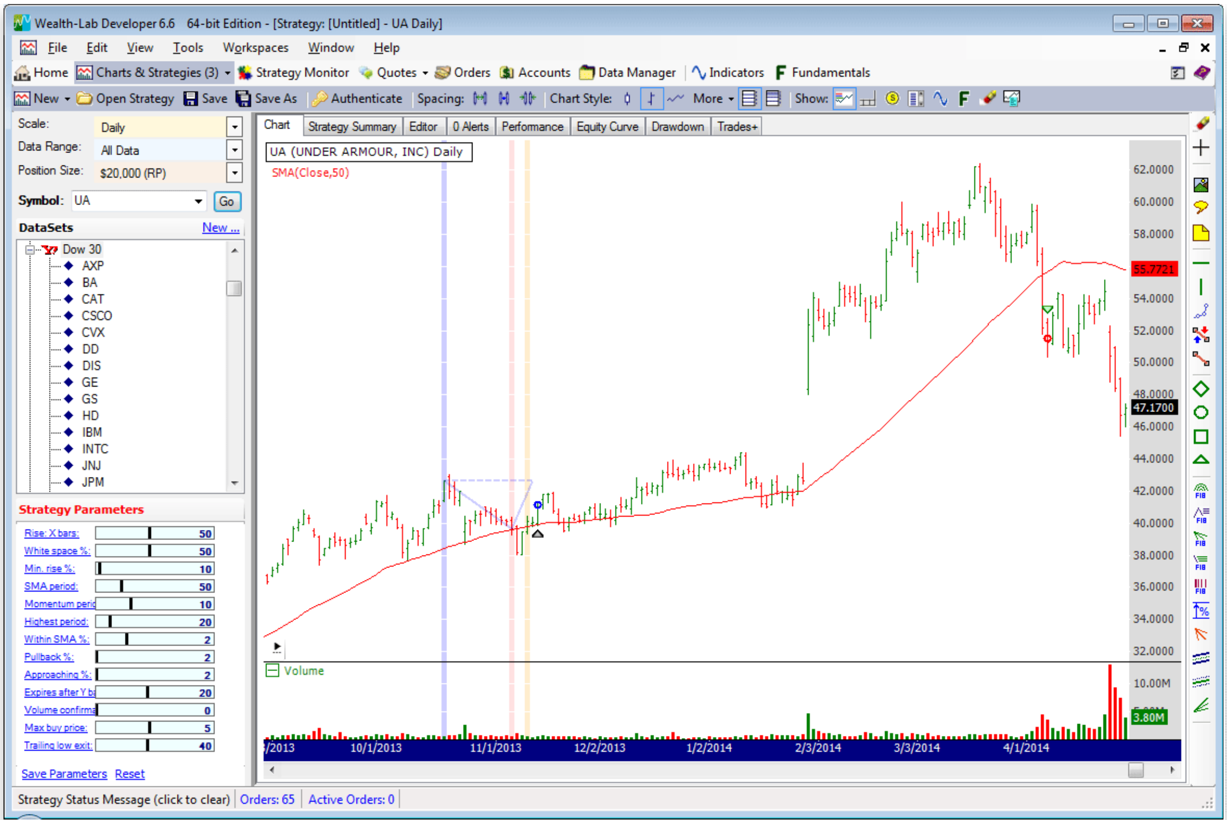
BACK TO LIST

WEALTH-LAB: SEPTEMBER 2014

In Charlotte Hudgin's article in this issue ("Finding The Golden Triangle"), she presents a formalized approach to identifying dips in stocks with established trends. Despite its clarity, the abundance of included system parameters (which includes various timeouts, periods, and percentages) makes it quite picky. You might want to reduce them if you notice that it catches too few setups, or to apply it to a large watchlist (think hundreds or even thousands of stocks) if you're an active trader.

Hudgin's idea of using "white space" between the price and its moving average is a clever visual technique to help find consistent trends that don't retrace too often and with higher momentum. We quantify the white space as a percentage of bars with their low price above a moving average. A potential enhancement might be an evaluation of *how much* the low price deviates from the SMA.

Unfortunately, as frequently happens, the article does not focus on getting *out* of the trade. We will leave development of a complete exit strategy to motivated traders, though a simple trailing exit that you will find in the code works pretty well. (See Figure 4.)



Wealth-Lab Developer 6.6 64-bit Edition - [Strategy: [Untitled] - UA Daily]

Scale: Daily
Data Range: All Data
Position Size: $20,000 (RP)
Symbol: UA

Strategy Parameters

Rise: X bars:	50
White space %:	50
Min. rise %:	10
SMA period:	50
Momentum perio	10
Highest period:	20
Within SMA %:	2
Pullback %:	2
Approaching %:	2
Expires after Y ba	20
Volume confirma	0
Max buy price:	5
Trailing low exit:	40

Strategy Status Message (click to clear) | Orders: 65 | Active Orders: 0

FIGURE 4: WEALTH-LAB. Here's a sample Wealth-Lab 6 chart illustrating the application of the system's rules on a daily chart of Under Armour Inc. (UA).

To execute the trading system, Wealth-Lab users may copy/paste our provided C# code shown below, or simply let Wealth-Lab do that job: in the *open strategy* dialog, click *download* to get the strategy code.

Spotting trend pullbacks is a very popular tactic. If you want more on this, Wealth-Lab offers variations of the same tune. In particular, we'd like to highlight another downloadable strategy, the "big move, pullback and continuation" system, which can be found under the *pullbacks* folder in the *open strategy* dialog (Ctrl-O). It also features great flexibility in configuring various parameters interactively through the use of "sliders."

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 GoldenTriangleStrategy : WealthScript
        {
                private StrategyParameter paramRiseBars;
                private StrategyParameter paramWhiteSpace;
                private StrategyParameter paramMinRise;
                private StrategyParameter paramSMA;
                private StrategyParameter paramMom;
                private StrategyParameter paramHi;
                private StrategyParameter paramProximity;
                private StrategyParameter paramPullback;
                private StrategyParameter paramRecovery;
                private StrategyParameter paramTimeout;
                private StrategyParameter paramVolConf;
                private StrategyParameter paramMaxBuy;
                private StrategyParameter paramTrail;

                public GoldenTriangleStrategy()
                {
                        paramRiseBars = CreateParameter("Rise: X bars", 50, 10, 100,
10);
                        paramWhiteSpace = CreateParameter("White space %", 50, 10,
100, 10);
                        paramMinRise = CreateParameter("Min. rise %", 10, 5, 200, 5);
                        paramSMA = CreateParameter("SMA period", 50, 10, 200, 10);
                        paramMom = CreateParameter("Momentum period", 10, 2, 30, 2);
                        paramHi = CreateParameter("Highest period", 20, 10, 100, 10);
                        paramProximity = CreateParameter("Within SMA %", 2, 1, 5, 1);
                        paramPullback = CreateParameter("Pullback %", 2, 2, 18, 2);
                        paramRecovery = CreateParameter("Approaching %", 2, 2, 6, 2);
                        paramTimeout = CreateParameter("Expires after Y bars", 20, 5,
40, 5);
                        paramVolConf = CreateParameter("Volume confirmation?", 0, 0,
1, 1);
                        paramMaxBuy = CreateParameter("Max buy price", 5, 1, 10, 1);
                        paramTrail = CreateParameter("Trailing low exit", 40, 10, 80,
10);
                }

                protected override void Execute()
                {
                        bool pivot = false; int pivotBar = -1;
                        bool pullback = false; int pullbackBar = -1;
                        bool recovery = false; int recoveryBar = -1;
                        bool volConfirm = paramVolConf.ValueInt == 1;
                        double ws = paramWhiteSpace.Value / 100d;
                        double within = paramProximity.Value;
                        double minRise = paramMinRise.Value;
                        double risePct = 0.0, pivotPrice = 0.0, dipPrice = 0.0;
                        int riseBars = paramRiseBars.ValueInt, ba = 0;

                        SMA sma = SMA.Series(Close,paramSMA.ValueInt);
                        MomentumPct mom = MomentumPct.Series(Close,paramMom.ValueInt);
                        Highest hi = Highest.Series(High,paramHi.ValueInt);
                        DataSeries whiteSpace = new DataSeries(Bars,"WhiteSpace");

                        Color blue = Color.FromArgb(50,Color.Blue);
                        LineStyle ls = LineStyle.Solid;
                        PlotSeries(PricePane,sma,Color.Red,ls,1);

                        for(int bar = Math.Max(riseBars,
GetTradingLoopStartBar(paramSMA.ValueInt)); bar < Bars.Count; bar++)
                        {
                                // "White space": percentage of bars above 50-day SMA
                                for (int i = bar - riseBars; i <= bar; i++)
                                {
                                        ba = (Low[i] > sma[i]) ? ba += 1 : 0;
                                        whiteSpace[bar] = ba / (double)riseBars;

}

                                if (IsLastPositionActive)
                                {
                                        SellAtStop( bar+1, LastPosition,
Lowest.Series(Low,paramTrail.ValueInt)[bar] );
                                }
                                else
                                {
                                        // 1. Detecting pivot
                                        if( !pivot )
                                        {
                                                // Uptrend: price > SMA, momentum % >
100, "white space" at or exceeds 50%, hit new 50-day high
                                                if( mom[bar] >= 100 &&
whiteSpace[bar] > ws && High[bar] >= hi[bar] )
                                                {
                                                        // Rise over X bars (default)
                                                        risePct = (High[bar] -
High[bar - riseBars]) / High[bar] * 100.0;

                                                        // Pivot detected: price rise
exceeds predefined % threshold
                                                        if( risePct > minRise )
                                                        {
                                                                pivot = true;
pivotBar = bar; pivotPrice = Close[pivotBar];
                                                                SetBackgroundColor(
pivotBar, blue );

}
                                                }
                                        }

                                        // 2. Looking for pullback
                                        if( pivot )
                                        {
                                                // Pullback is valid until it times
out
                                                if( bar <= pivotBar +
paramTimeout.ValueInt )

{
                                                        if( !pullback )
                                                        {
                                                                // Pullback detected:
price dove within N% of SMA
                                                                bool priceNearSMA =
Close[bar] > (sma[bar] * 1 - (within / 100d)) &&
                                                                        Close[bar] <
(sma[bar] * 1 + (within / 100d));
                                                                if( priceNearSMA )
                                                                {
                                                                        pullback =
true; pullbackBar = bar; dipPrice = Close[pullbackBar];
```

```
SetBackgroundColor( pullbackBar, Color.FromArgb(30, Color.Red) );
                                                                }
                                                        }

                                                        // 3. Looking for recovery
                                                        if( pullback )
                                                        {
                                                                // Rebound is valid
until it times out
                                                                if( bar <=
pullbackBar + paramTimeout.ValueInt )
                                                                {
                                                                        if( !recovery
)
                                                                        {
                                                                                //
Recovery started: current price is above both the 50-day SMA and Pullback price
                                                                                //
but current high is still below the Pivot price
                                                                                if(
(Close[bar] > sma[bar]) && (Close[bar] > dipPrice) && (High[bar] <= High[pivotBar]) )
                                                                                {

        recovery = true; recoveryBar = bar;

        SetBackgroundColor( recoveryBar, Color.FromArgb(50, Color.Orange) );
                                                                                }
                                                                        }


                                                                        // 4. Looking
to enter
                                                                        if( recovery )
                                                                        {
                                                                                //
4.a Price confirmation
                                                                                if(
Close[bar] > sma[bar] )
                                                                                {

        // 4.b Volume confirmation (if enabled)

        if( !volConfirm || (volConfirm && Volume[bar] > SMA.Series(Volume,50)[bar]) )

        {

                // Enter: price below Max Buy price

                double maxBuyPrice = (1 - paramMaxBuy.Value / 100d) * High[pivotBar];

                if( Close[bar] < maxBuyPrice )

                {

                        // Eugene: buy at stop half-way between the Pivot and
Pullback prices (or higher)

                        if( BuyAtStop( bar + 1, (dipPrice + (pivotPrice - dipPrice) /
2), Bars.FormatValue(risePct) ) != null )
```

```
                        // Alternative:

                        //if( BuyAtMarket( bar + 1, Bars.FormatValue(risePct) ) !=
null )

                        {

                                DrawLine( PricePane, pivotBar, pivotPrice,
pullbackBar, dipPrice, blue, ls, 2 );

                                DrawLine( PricePane, pullbackBar, dipPrice, bar,
High[pivotBar], blue, ls, 2 );

                                DrawLine( PricePane, pivotBar, High[pivotBar], bar,
High[pivotBar], blue, LineStyle.Dashed, 2 );



                                pivot = false; pullback = false; recovery = false;

                                LastPosition.Priority = -Close[bar];

                        }

                        else

                                // reset if setup has timed out

                                recovery = bar + 1 - recoveryBar <
paramTimeout.ValueInt;

                }

        }
                                                                                }
                                                                        }
                                                                }
                                                                else
                                                                {
                                                                        pullback =
false;

SetBackgroundColor( pullbackBar, Color.Transparent );
                                                                }

}
                                                }
                                                else
                                                {
                                                        pivot = false;
                                                        SetBackgroundColor( pivotBar,
Color.Transparent );
                                                }
                                        }
                                }
                        }
                }
        }
}
```

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

BACK TO LIST

NEUROSHELL TRADER: SEPTEMBER 2014

A trading system based on the setup for the Armchair Investor Golden Triangle presented by Charlotte Hudgin in her article in this issue ("Finding The Golden Triangle") can be easily implemented in NeuroShell Trader using a few of the many indicators our program offers. Simply select "New trading strategy" from the Insert menu and enter the following in the appropriate locations of the trading strategy wizard:

```
Generate a buy long Market order if all of the following are true:

A>B(Lag(Max(A>B(LinTimeReg Slope(Close,50),LinTimeReg
Slope(Avg(Close,50),50)),20),2),0)

And3(Or2(And2(CrossBelow(Low,Avg(Close,50)),A>B(Close,Avg(Close,50))),And2(A>B(Max(Cro
ssBelow(Low,Avg(Close,50)),10),0),A>B(Momentum(Close,1),0))),A>B(Volume,Avg(Volume,50)
),High Channel Breakout(Volume,5))

Protective Stop:

     TrailPrice%(Trading Strategy,10)
```

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 website to download a copy of this or any previous Traders' Tips.

A sample chart is shown in Figure 5.

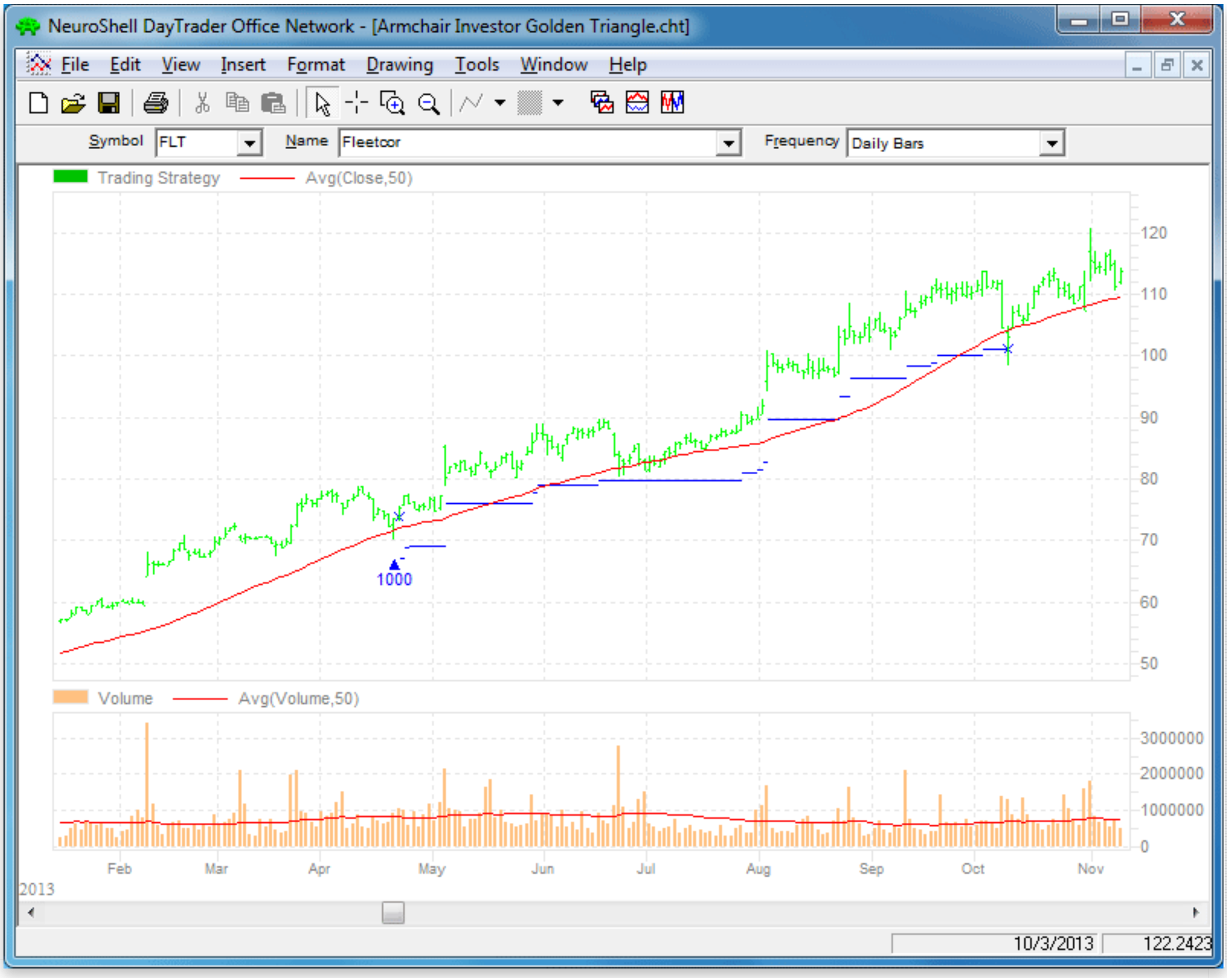


FIGURE 5: NEUROSHELL TRADER. This NeuroShell Trader chart displays the Armchair Investor Golden Triangle entry followed by a trailing-stop exit.

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

BACK TO LIST

NINJATRADER: SEPTEMBER 2014

We have implemented the golden triangle strategy that is presented by Charlotte Hudgin in her article in this issue, “Finding The Golden Triangle,” and we have made it available for download at www.ninjatrader.com/SC/September2014SC.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 indicator source code by selecting the menu Tools → Edit NinjaScript → Strategy from within the NinjaTrader Control Center window and selecting the “GoldenTriangle” file.

A sample chart implementing the strategy is shown in Figure 6.

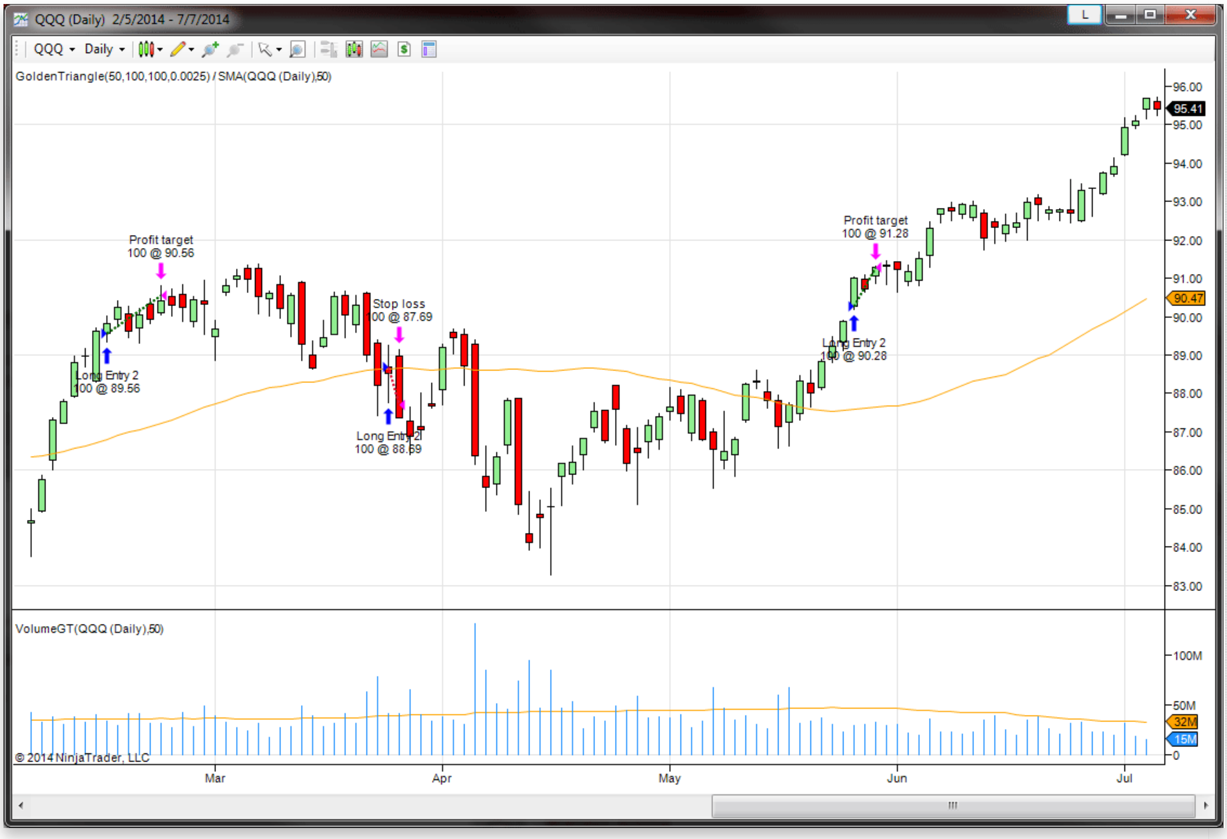


FIGURE 6: NINJATRADER. This screenshot shows the strategy applied to a daily QQQ chart in NinjaTrader.

—Raymond Deux and Cal Hueber
NinjaTrader, LLC
www.ninjatrader.com

BACK TO LIST

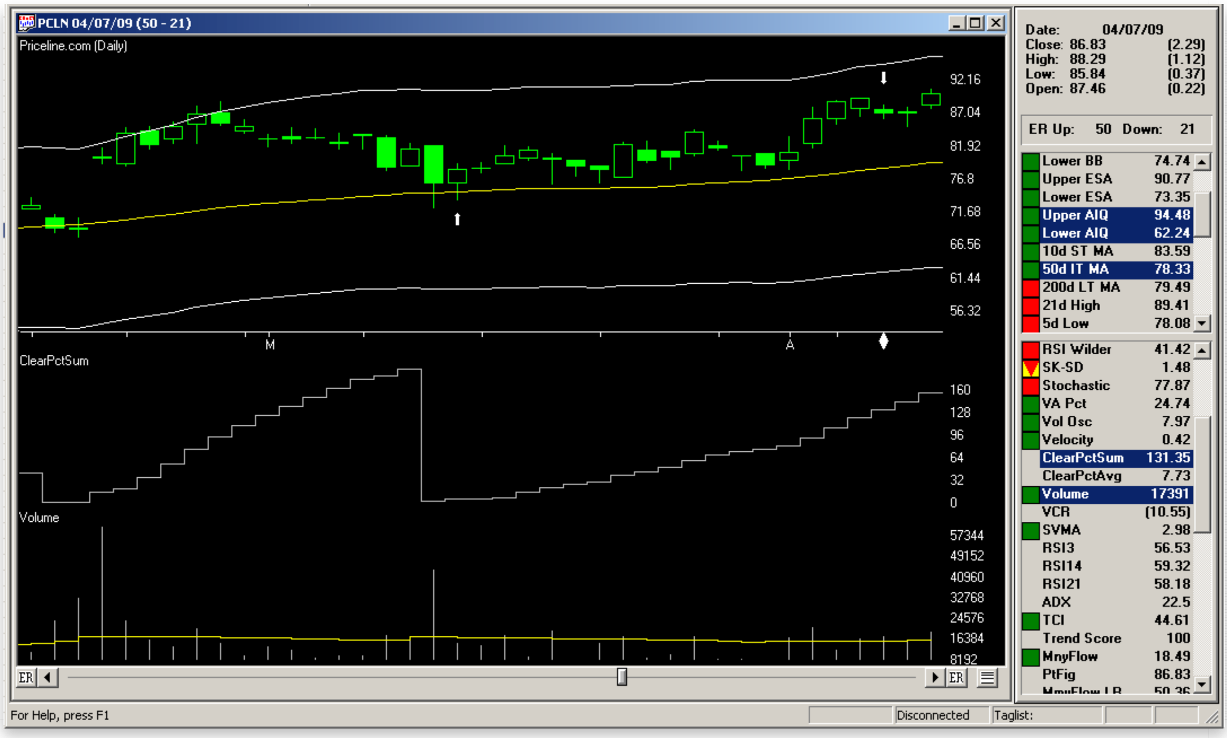
AIQ: SEPTEMBER 2014

The AIQ code and EDS file based on Charlotte Hudgin’s article in this issue, “Finding The

Golden Triangle," is provided at www.TradersEdgeSystems.com/traderstips.htm, and is shown below.

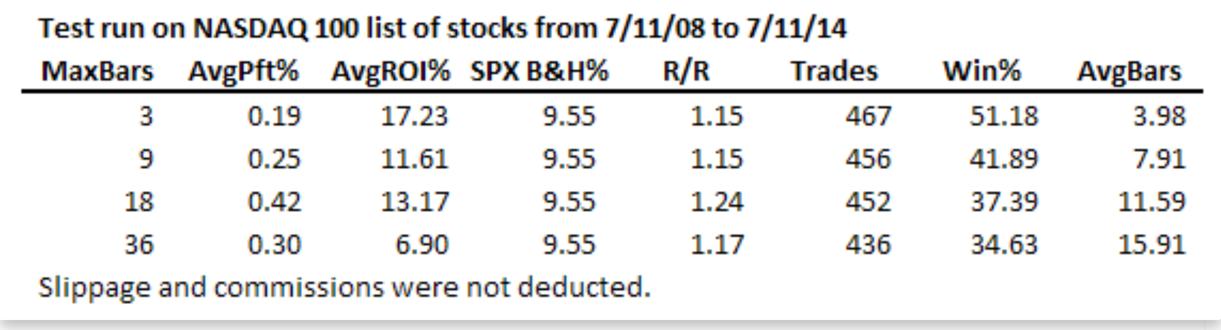
I created an indicator I named the *clear value indicator* ("ClearValueSum" and "ClearValueAvg") that might be used to rank signals. The "ClearValueSum" indicator sums the daily percentages that the close is above the simple moving average (SMA). The summing starts at the last cross up and goes to the current bar. If the close is below the SMA, then the value of the indicator is zero. In Figure 7, I show a chart of Priceline (PCLN) with the ClearVauleSum indicator in the subgraph. In addition, I provide the code for the golden triangle setup and confirmation.

The author did not discuss exits, so I provided one based on a cross under the SMA or an exit after a maximum-bars-to-hold input ("maxBarsToHold").



```
Date:        04/07/09
Close:  86.83     (2.29)
High:   88.29     (1.12)
Low:    85.84     (0.37)
Open:   87.46     (0.22)

ER Up:   50   Down:   21

🟩 Lower BB       74.74
🟩 Upper ESA      90.77
🟩 Lower ESA      73.35
🟩 Upper AIQ      94.48
🟩 Lower AIQ      62.24
🟩 10d ST MA      83.59
🟩 50d IT MA      78.33
🟥 200d LT MA     79.49
🟥 21d High       89.41
🟥 5d Low         78.08

🟥 RSI Wilder     41.42
▼ SK-SD           1.48
🟥 Stochastic     77.87
🟩 VA Pct         24.74
🟩 Vol Osc         7.97
🟩 Velocity        0.42
   ClearPctSum   131.35
   ClearPctAvg     7.73
🟩 Volume         17391
   VCR           (10.55)
🟩 SVMA            2.98
   RSI3           56.53
   RSI14          59.32
   RSI21          58.18
   ADX            22.5
🟩 TCI            44.61
   Trend Score    100
🟩 MnyFlow        18.49
   PtFig          86.83
   MnyFlow LB     50.36
```

FIGURE 7: AIQ, sample trade. Here is a chart of Priceline (PCLN) with the ClearValueSum indicator and a sample trade marked with white up and down arrows.



Test run on NASDAQ 100 list of stocks from 7/11/08 to 7/11/14

MaxBars	AvgPft%	AvgROI%	SPX B&H%	R/R	Trades	Win%	AvgBars
3	0.19	17.23	9.55	1.15	467	51.18	3.98
9	0.25	11.61	9.55	1.15	456	41.89	7.91
18	0.42	13.17	9.55	1.24	452	37.39	11.59
36	0.30	6.90	9.55	1.17	436	34.63	15.91

Slippage and commissions were not deducted.

FIGURE 8: AIQ, SAMPLE PERFORMANCE RESULTS. Here are the EDS summary results compared with varying the maxBarsToHold input trading the NASDAQ 100 list of stocks over the last six years.

I ran a short optimization on the “maxBarsToHold” input, the results of which are shown in the table in Figure 8. Most of the metrics are best at the 18-bar setting. In Figure 7, I also show a sample trade from the system from 2009 with the 18-bar setting.

```
!FINDING THE GOLDEN TRIANGLE
!Author: Charlotte Hudgin, TASC Sept 2014
!Coded by: Richard Denning 7/10/2014
!www.TradersEdgeSystems.com

!INPUTS:
        smaLen is 50.        !moving average length
        periods is 252.      !Total look back period
        strength is 4.       !Number of bars on each side of pivot
        maxBarsToHold is 18. !max bars to hold position

!VARIABLES:
        C is [close].
        L is [low].
        V is [volume].
        OTD is offsettodate(month(),day(),year()).

!CLEAR VALUE INDICATOR:
        SMA is simpleavg(C,smaLen).
        Xup if C>SMA and (valrule(C<=SMA,1) or countof(L<SMA,2)>=1).
        XupDte is scanany(Xup,periods).
        XupOS is scanany(Xup,periods) then OTD.
        ClearPct is (C/SMA -1) * 100.
        ClearPctSum is iff(C>SMA,sum(ClearPct,^XupOS),0).
        ClearPctAvg is iff(C>SMA and
^XupOS>1,simpleavg(ClearPct,^XupOS),iff(ClearPct>0,ClearPct,0)).

!CODE TO FIND PIVOTS:
        LowR is LoVal([low],(2*strength)+1).
        LowM is Val([low],strength).
        LS if LowR = LowM.
        HighR is HiVal([high],(2*strength)+1).
        HighM is Val([high],strength).
        HS if  HighR = HighM.

   !FIND FIRST PIVOT LOW
        LT1 is scanany(LS,periods) then OTD .
        LO1 is ^LT1 + Strength.
        LO1dte is SetDate(LO1).
        LowLO1 is val([low],^LO1).

   !FIND FIRST PIVOT HIGH
        HT1 is scanany(HS,periods,0) then OTD .
        HO1 is ^HT1 + Strength.
        HO1dte is SetDate(HO1).
        HighHO1 is val([high],HO1).
```

```
!SYSTEM CODE:
        Xdn if [low]<SMA and valrule([low]>=SMA,1).
        XdnDte is scanany(Xdn,periods).
        XdnOS is scanany(Xdn,periods) then OTD.

        ShowValues if C > 5.
        HHVpivot if HighHO1 = hival([high],smaLen) and C > 5.
        Setup if Xdn and HHVpivot.
        PriceCnf if C>SMA.
        SetupOS is scanany(Setup,periods) then OTD.
        PriceCnfOS is scanany(PriceCnf,periods) then OTD.
        AvgV is simpleavg(V,smaLen).
        VolumeCnf if ^SetupOS<15 and SetupOS<=^PriceCnfOS and V>avgV and
V=highresult(V,^PriceCnfOS).

    !BUY & EXIT RULES (LONG ONLY):
        Buy if VolumeCnf and countof(Setup,15)=1 and countof(PriceCnf,15)>=1
                and countof(C>SMA,SetupOS+1)=SetupOS+1.
        Exit if C<SMA or  {position days}>=maxBarsToHold.
```

—Richard Denning
info@TradersEdgeSystems.com
for AIQ Systems

BACK TO LIST

TRADERSSTUDIO: SEPTEMBER 2014

The TradersStudio code based on Charlotte Hudgin's article in this issue, "Finding The Golden Triangle," is provided at both of the following websites:

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

The following code file is provided in the download:

- System GOLDEN_TRIANGLE: A trading system described by Charlotte Hudgin using for the exit a max-bars-held input

The system has the following rules:

- Buy at the next bar at market open after the setup, price confirmation, and volume confirmation occur within 15 bars
- Exit the long position at the next bar at market open when the close crosses under the SMA
- Exit the long position on the next bar at market open when the max bars held have been reached
- As coded, this is a long-only system.

I ran the system on a sample futures portfolio using data from Pinnacle Data Corp. (www.pinnacledata.com). Figure 9 shows the equity and underwater equity curves trading one contract for each signal. The test period was from the year 2000 to 2014. The table in Figure 10 shows the session setup parameters and the list of symbols in the portfolio.

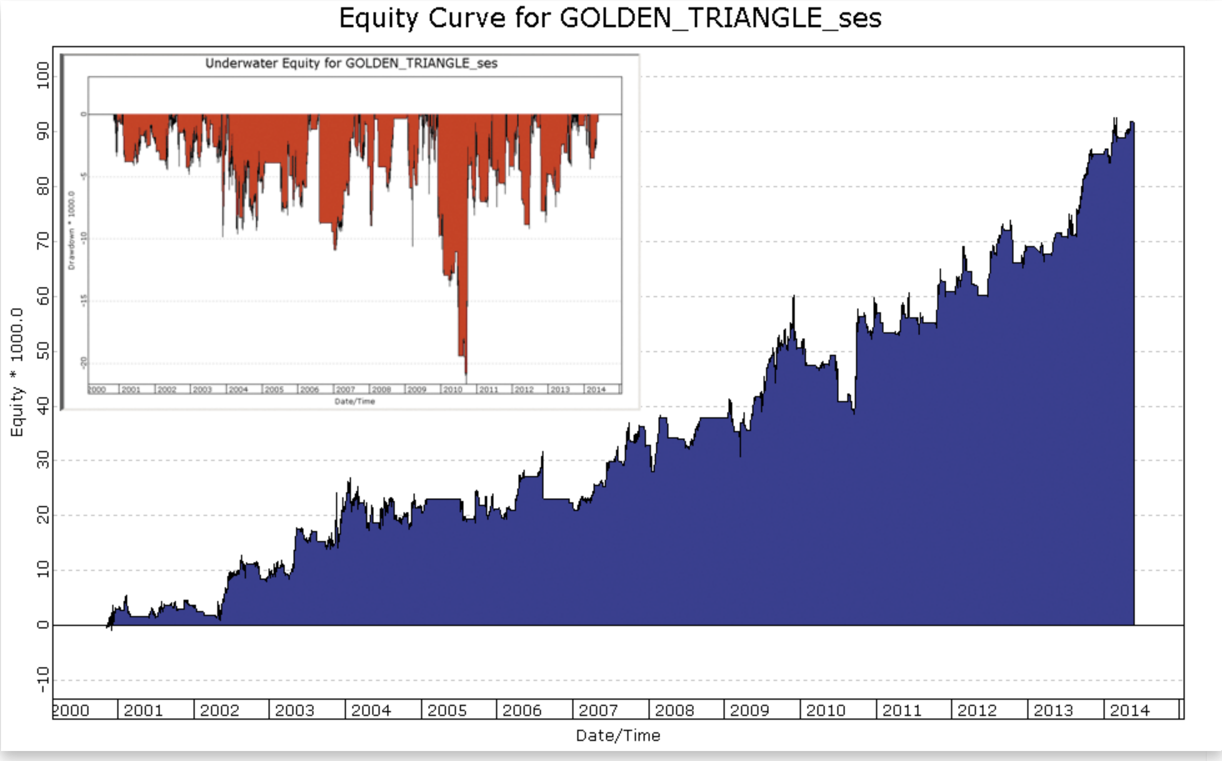
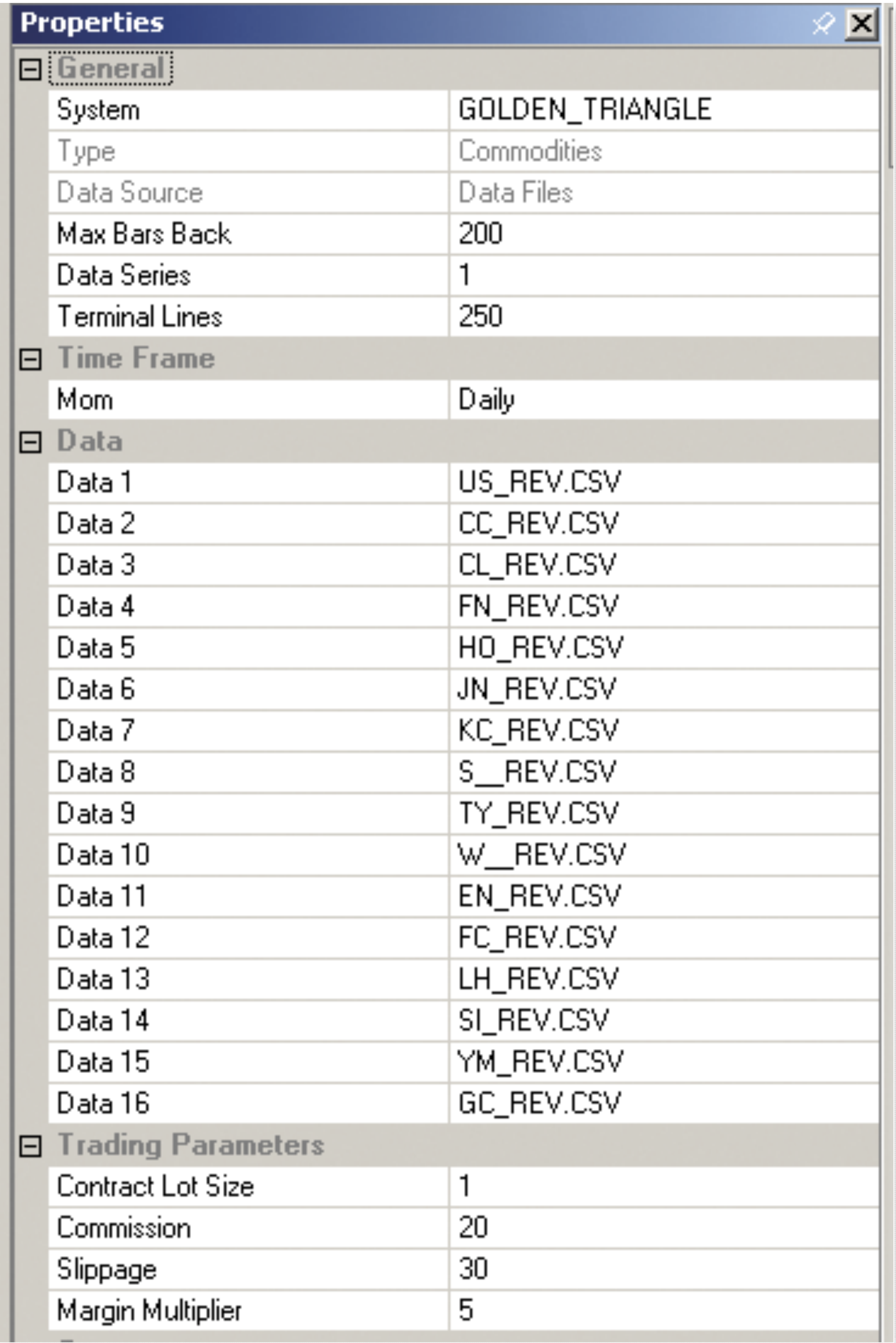


FIGURE 9: TRADERSSTUDIO, EQUITY CURVE. Here are sample equity and underwater equity curves trading one contract for each signal on a portfolio of futures contracts for the period 2000–2014.



⊟ General	
System	GOLDEN_TRIANGLE
Type	Commodities
Data Source	Data Files
Max Bars Back	200
Data Series	1
Terminal Lines	250
⊟ Time Frame	
Mom	Daily
⊟ Data	
Data 1	US_REV.CSV
Data 2	CC_REV.CSV
Data 3	CL_REV.CSV
Data 4	FN_REV.CSV
Data 5	HO_REV.CSV
Data 6	JN_REV.CSV
Data 7	KC_REV.CSV
Data 8	S__REV.CSV
Data 9	TY_REV.CSV
Data 10	W__REV.CSV
Data 11	EN_REV.CSV
Data 12	FC_REV.CSV
Data 13	LH_REV.CSV
Data 14	SI_REV.CSV
Data 15	YM_REV.CSV
Data 16	GC_REV.CSV
⊟ Trading Parameters	
Contract Lot Size	1
Commission	20
Slippage	30
Margin Multiplier	5

FIGURE 10: TRADERSSTUDIO. Here are the session

setup parameters and the list of symbols in the portfolio used for the test run.

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

UPDATA: SEPTEMBER 2014

Our Traders' Tips this month is based on the article in this issue by Charlotte Hudgin, "Finding The Golden Triangle."

In the article, Hudgin refines the "dip-buying" concept by introducing a triangle-pattern search within the underlying data. According to this concept, fast-growing stocks can experience some pullbacks before continuing an overall uptrend. Analysis of price and volume action during the pullback tries to anticipate those instruments likely to return to an uptrend the earliest.

The Updata code based on Hudgin's article has been added to the Updata Library and may be downloaded by clicking the *custom* menu and then *indicator library*. The code is also shown below for pasting into the Updata custom editor. A sample chart implementing the strategy is shown in Figure 11.

```
'Golden Triangle
DISPLAYSTYLE 4LINES
INDICATORTYPE TOOL
INDICATORTYPE3 CHART
PLOTSTYLE3 HISTOGRAM
COLOUR2 RGB(0,0,200)
COLOUR4 RGB(0,0,200)
NAME "Golden Triangle" ""
NAME3 "Mov Avg" ""
PARAMETER "Period B-C" #PeriodBC=9
PARAMETER "Period A-B" #PeriodABB=14
PARAMETER "Avg. Period" #AvgPeriod=50
@Avg=0
@DownCondition=0
@UpCondition=0
#INC=0
#i=0
#PeriodAB=0
@VolumeAvg=0

FOR #CURDATE=#AvgPeriod TO #LASTDATE
```

```
    @Avg=MAVE(#AvgPeriod)
    @VolumeAvg=SGNL(VOL,#AvgPeriod,M)
    @DownCondition=0
    @UpCondition=0
    #PeriodAB=#PeriodABB
    For #i=0 TO #PeriodAB-1
        if CLOSE(#PeriodBC+#i)<CLOSE(#PeriodBC+#i+1) AND CLOSE(#PeriodBC+#i)
<CLOSE(#PeriodBC+#PeriodAB)
           @DownCondition=@DownCondition+1
        endif
        if #i<#PeriodBC
           if CLOSE(#i)>CLOSE(#i+1) AND CLOSE(#i)>CLOSE(#PeriodAB)
              @UpCondition=@UpCondition+1
           endif
        endif
        if CLOSE(#PeriodBC+#i)<CLOSE(#PeriodBC+#i+1) AND CLOSE(#PeriodBC+#i)
<CLOSE(#PeriodBC+#PeriodAB) AND #i=#PeriodAB-1
           #PeriodAB=#PeriodAB+1
        endif
    Next
    If @UpCondition>0.65*#PeriodBC AND @DownCondition>0.65*#PeriodAB
       #INC=#INC+1
        DRAWLINE
(#PeriodBC+#PeriodAB),CLOSE(#PeriodBC+#PeriodAB),#PeriodBC,CLOSE(#PeriodBC)
        DRAWLINE #PeriodBC,CLOSE(#PeriodBC),0,CLOSE(0)
    EndIf
    @PLOT2=@Avg
    @PLOT3=VOL
    @PLOT4=@VolumeAvg
NEXT
```

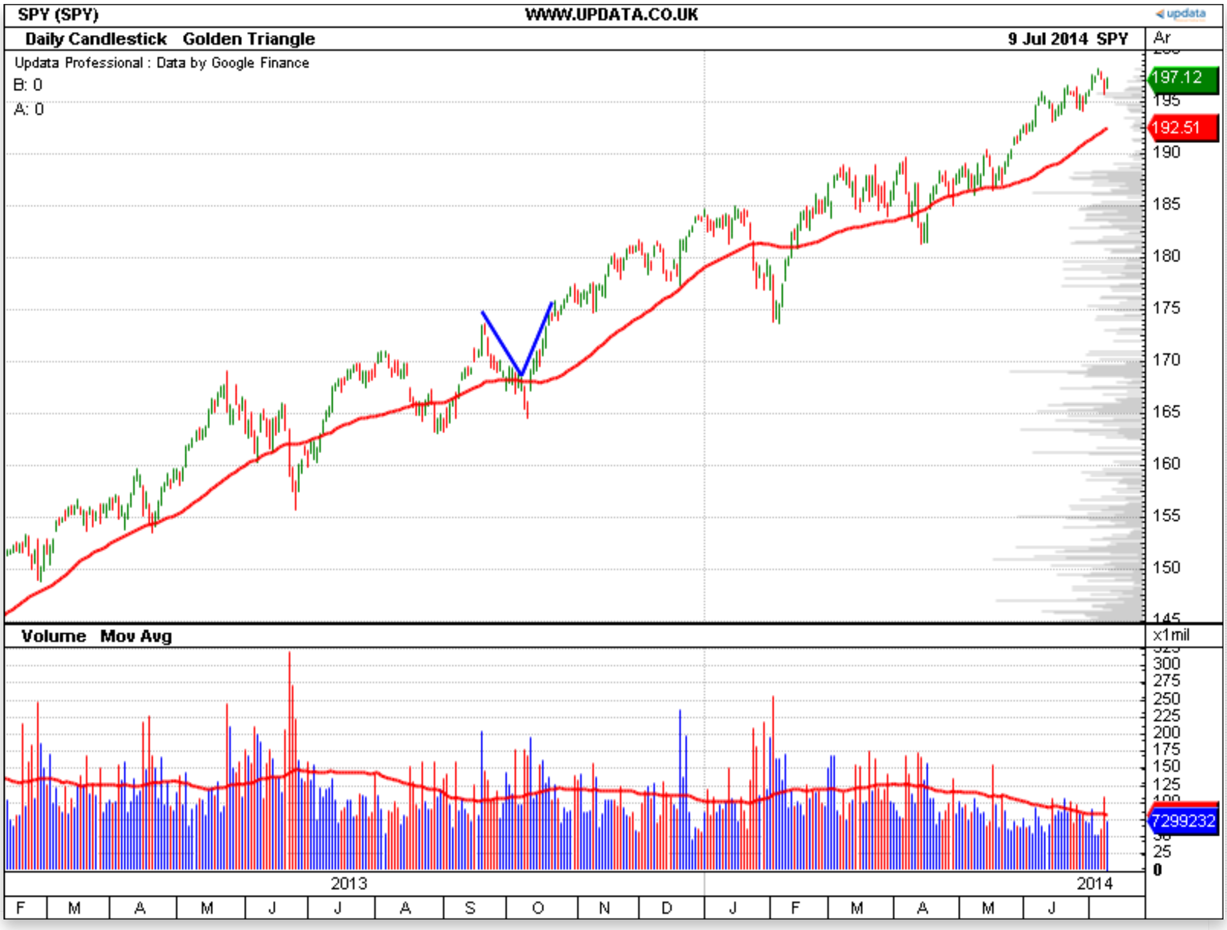


FIGURE 11: UPDATA. This sample chart shows the golden triangle as applied to the SPY ETF in daily resolution.

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

BACK TO LIST

October 2014

For this month's Traders' Tips, the focus is mainly Sylvain Vervoort's article from the September 2014 issue, "Exploring Charting Techniques: Creating A Trading Strategy, Part 3." Here we present the October 2014 Traders' Tips code with possible implementations in various software.

Code for NinjaTrader was already provided with Vervoort's article by the author. S&C subscribers will find that code at the Subscriber Area of our website here. (Click on "S&C Article Code" from the homepage.) 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: OCTOBER 2014

In "Exploring Charting Techniques: Creating A Trading Strategy, Part 3," which appeared last month in the September 2014 issue of STOCKS & COMMODITIES, author Sylvain Vervoort describes a process for creating indicators and strategies. Through his article series, the author takes us step by step, building up his trading idea with additional criteria along the way.

We are providing EasyLanguage code for both upper (moving averages) and lower (trading signal) indicators, as well as a strategy based on the author's ideas (see Figure 1). As Vervoort points out, advanced chart types can be automated only when the limitations are fully understood.

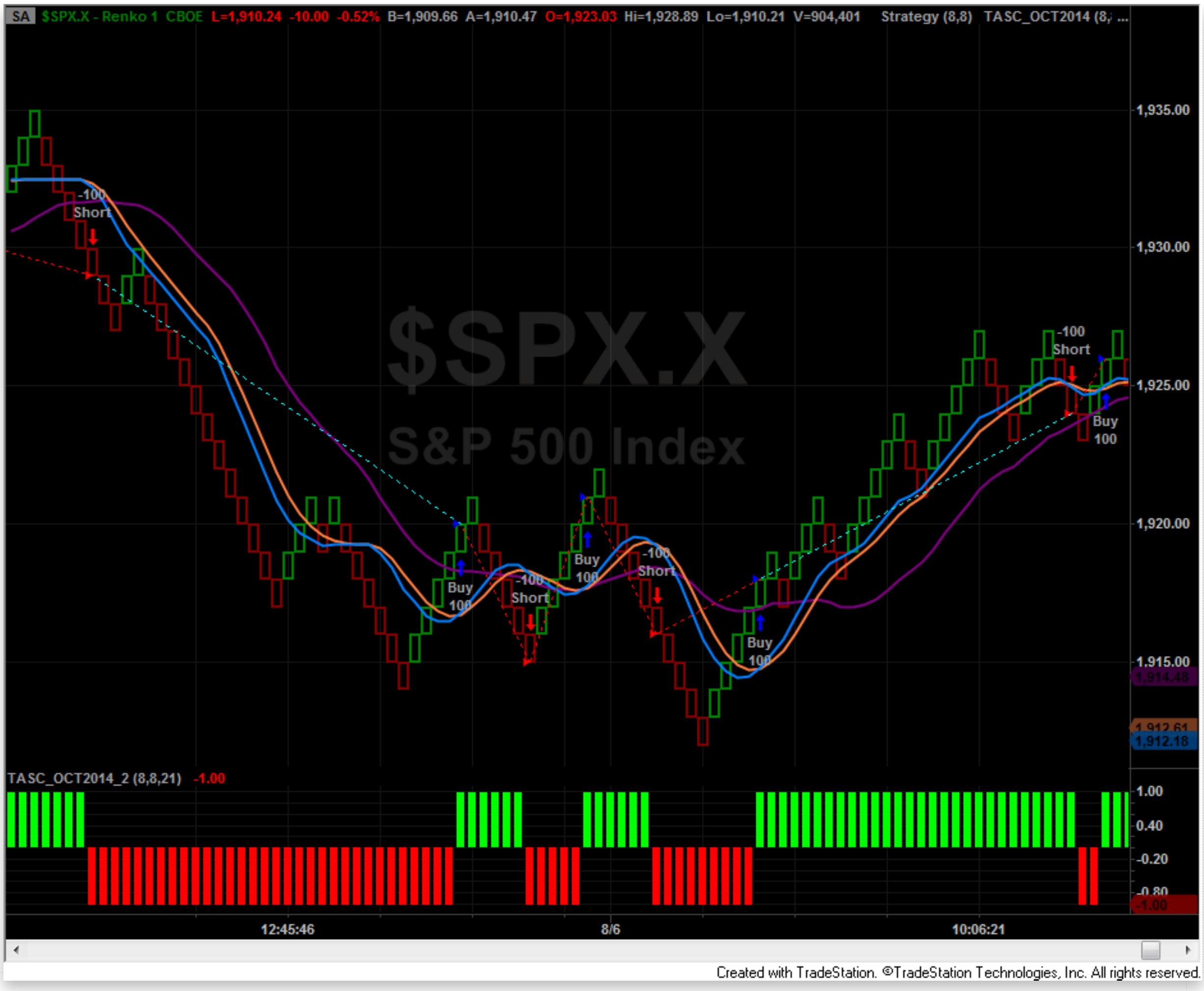

SA $SPX.X - Renko 1 CBOE L=1,910.24 -10.00 -0.52% B=1,909.66 A=1,910.47 O=1,923.03 Hi=1,928.89 Lo=1,910.21 V=904,401 Strategy (8,8) TASC_OCT2014 (8,; ...
$SPX.X
S&P 500 Index
-100 Short
Buy 100
-100 Short
Buy 100
-100 Short
Buy 100
-100 Short
Buy 100
1,935.00
1,930.00
1,925.00
1,920.00
1,915.00
1,914.48
1,912.61
1,912.18
TASC_OCT2014_2 (8,8,21) -1.00
1.00
0.40
-0.20
-0.80
-1.00
12:45:46
8/6
10:06:21
Created with TradeStation. ©TradeStation Technologies, Inc. All rights reserved.

FIGURE 1: TRADESTATION. Here is an example implementation of the strategy and indicators based on author Sylvain Vervoort's ideas applied to a chart of the S&P 500 index using a one-point renko bar.

More on strategy backtesting and automation of TradeStation's range, renko, and other advanced chart types can be found in the *advance chart type* entry within *platform help* (from the TradeStation *platform* help menu, select *platform help*).

To download the EasyLanguage code for TradeStation, please visit our TradeStation and EasyLanguage support forum. The code from this article can be found here: http://www.tradestation.com/TASC-2014. The ELD filename is "_TASC_OCT2014_CREATINGSTRATEGY.ELD."

```
TASC_Oct2014 (Strategy)

inputs:
TypicalPriceAvgLength( 8 ),
HeikinAshiAvgLength( 8 ),
TrendAvgLength( 21 ) ;

variables:
        haClose( 0 ),
        haOpen( 0 ),
        haHigh( 0 ),
        haLow( 0 ),
        HAAverage( 0 ),
        TPAverage( 0 ),
        TrendAverage( 0 ),
        TrendValue( 0 ),
        SignalCondition( false ),
        BarCondition( false ),
        TrendCondition( false ) ;

haClose = ( Open + High + Low + Close ) / 4 ;
haOpen = ( haOpen[1] + haClose[1] ) / 2 ;
haHigh = MaxList( High, haOpen, haClose ) ;
haLow = MinList( Low, haOpen, haClose ) ;

HAAverage = Average( ( haClose + haOpen + haHigh + haLow )
        / 4, HeikinAshiAvgLength ) ;
TPAverage = Average( TypicalPrice,
        TypicalPriceAvgLength ) ;
TrendAverage = Average( Close, TrendAvgLength ) ;

SignalCondition = TPAverage > HAAverage ;
BarCondition = Close > Open ;
TrendCondition = Close > TrendAverage ;

if SignalCondition and BarCondition
        and TrendCondition then
        TrendValue = 1
else if not SignalCondition and
        not BarCondition and not TrendCondition then
        TrendValue = -1
else
        TrendValue = TrendValue[1] ;

if TrendValue = 1 then
        Buy ("LE") this bar on Close
else if TrendValue = -1 then
        SellShort ("SE") this bar on Close ;

if TPAverage crosses under HAAverage then
        Sell ("LX")this bar on Close
else if TPAverage crosses over HAAverage then
        BuyToCover ("SX") this bar on Close ;
```

TASC_Oct2014 (Indicator)

```
inputs:
TypicalPriceAvgLength( 8 ),
HeikinAshiAvgLength( 8 ),
TrendAvgLength( 21 ) ;

variables:
        haClose( 0 ),
        haOpen( 0 ),
        haHigh( 0 ),
        haLow( 0 ),
        HAAverage( 0 ),
        TPAverage( 0 ),
        TrendAverage( 0 ),
        TrendValue( 0 ),
        PlotColor( 0 ) ;

haClose = ( Open + High + Low + Close ) / 4 ;
haOpen = ( haOpen[1] + haClose[1] ) / 2 ;
haHigh = MaxList( High, haOpen, haClose ) ;
haLow = MinList( Low, haOpen, haClose ) ;

HAAverage = Average( ( haClose + haOpen + haHigh + haLow )
        / 4, HeikinAshiAvgLength ) ;
TPAverage = Average( TypicalPrice,
        TypicalPriceAvgLength ) ;
TrendAverage = Average( Close, TrendAvgLength ) ;

Plot1( TPAverage, "TP Avg" ) ;
Plot2( HAAverage, "HA Avg" ) ;
Plot3( TrendAverage, "TrendAverage" ) ;
```

TASC_Oct2014_2 (Indicator)

```
inputs:
TypicalPriceAvgLength( 8 ),
HeikinAshiAvgLength( 8 ),
TrendAvgLength( 21 ) ;

variables:
        haClose( 0 ),
        haOpen( 0 ),
        haHigh( 0 ),
        haLow( 0 ),
        HAAverage( 0 ),
        TPAverage( 0 ),
        TrendAverage( 0 ),
        TrendValue( 0 ),
        PlotColor( 0 ),
        SignalCondition( false ),
        BarCondition( false ),
        TrendCondition( false ) ;

haClose = ( Open + High + Low + Close ) / 4 ;
haOpen = ( haOpen[1] + haClose[1] ) / 2 ;
haHigh = MaxList( High, haOpen, haClose ) ;
haLow = MinList( Low, haOpen, haClose ) ;

HAAverage = Average( ( haClose + haOpen + haHigh + haLow )
        / 4, HeikinAshiAvgLength ) ;
TPAverage = Average( TypicalPrice,
        TypicalPriceAvgLength ) ;
TrendAverage = Average( Close, TrendAvgLength ) ;

SignalCondition = TPAverage > HAAverage ;
BarCondition = Close > Open ;
TrendCondition = Close > TrendAverage ;

if SignalCondition and BarCondition
        and TrendCondition then
        TrendValue = 1
```

```
  else if not SignalCondition and
          not BarCondition and not TrendCondition then
          TrendValue = -1
  else
          TrendValue = TrendValue[1] ;

  if TrendValue = 1 then
          PlotColor = Green
  else if TrendValue = -1 then
          PlotColor = Red
  else
          PlotColor = Transparent ;

  Plot1( TrendValue, "TrendValue", PlotColor ) ;
```

For more information about EasyLanguage in general, please see http://www.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: OCTOBER 2014

For this month’s Traders’ Tip, we’ve provided the formula SVEHaTypeCrossStrategy.efs based on the formula described in Sylvain Vervoort’s article in the September 2014 issue of STOCKS & COMMODITIES, “Exploring Charting Techniques: Creating a Trading Strategy, Part 3.”

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.

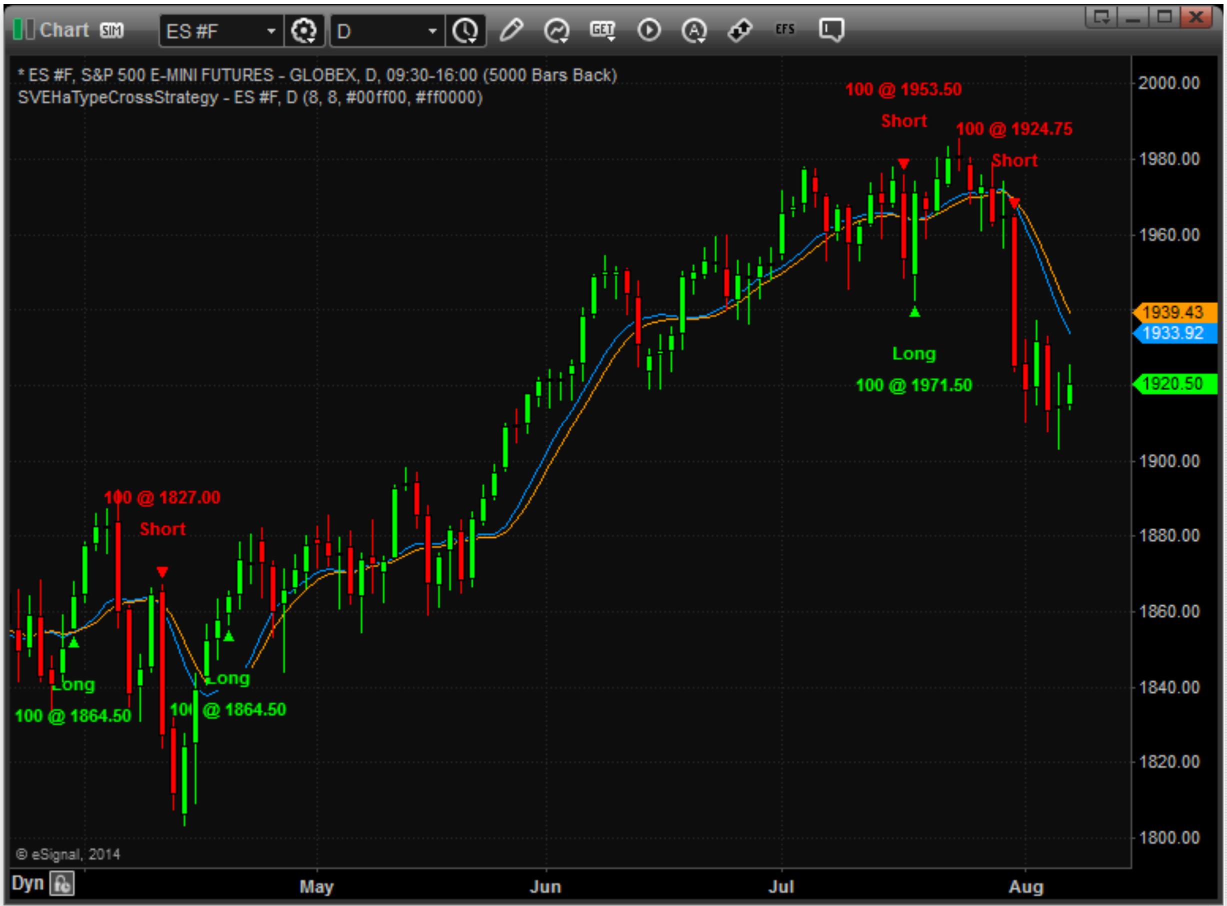


FIGURE 2: eSIGNAL. Here is an example of the SVEHaTypeCrossStrategy.efs shown on a chart of S&P 500 emini futures (ES).

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 for copying & pasting or for downloading here.

```
/**********************************
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:
    The Crossing Of Two Specific Moving Averages by Sylvain Vervoort

Formula Parameters:                     Default:
Typical Price Average                   8
Heikin-Ashi average                     8
Long Position Color                     lime
Short Position Color                    red
```

```
Version:            1.00  08/06/2014

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("SVEHaTypeCrossStrategy");
    setPriceStudy(true);

    setCursorLabelName("Typical Price Average", 0);
    setCursorLabelName("Heikin-Ashi Average", 1);

    setDefaultBarFgColor(Color.RGB(0x00,0x94,0xFF), 0);
    setDefaultBarFgColor(Color.RGB(0xFF,0x9B,0x00), 1);

    var x = 0;

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

        setName("Typical Price Average");
        setLowerLimit(1);
        setDefault(8);
    };

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

        setName("Heikin-Ashi Average");
        setLowerLimit(1);
        setDefault(8);
    };

    fpArray[x] = new FunctionParameter("fpLongColor", FunctionParameter.COLOR);
    with(fpArray[x++]){

        setName("Long Position Color");
        setDefault(Color.lime);
    };

    fpArray[x] = new FunctionParameter("fpShortColor", FunctionParameter.COLOR);
    with(fpArray[x++]){

        setName("Short Position Color");
        setDefault(Color.red);
    };
}

var bInit = false;
var bVersion = null;

var xClose = null;
var xOpen = null;
var xOHLC4 = null;
var xHLC3 = null;
var xHaOpen = null;
var xHaC = null;
var xAVGTyp = null;
var xAVGHaC = null;
var xCross = null;

function main(fpHaCAverage, fpTypicalAverage, fpLongColor, fpShortColor){

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

    if (!bInit){

        xClose = close();
        xOpen = open();
        xOHLC4 = ohlc4();
        xHLC3 = hlc3();

        xHaOpen = efsInternal("Calc_HaOpen", xOHLC4);
        xHaC = efsInternal("Calc_HaC", xOHLC4, xHaOpen);

        xAVGTyp = getSeries(sma(fpTypicalAverage, xHLC3));
        xAVGHaC = getSeries(sma(fpHaCAverage, xHaC));
        xCross = efsInternal("Calc_Cross", xAVGTyp, xAVGHaC, xClose, xOpen)

        bInit = true;
    };

    var nCross_Last = xCross.getValue(-1);
    var nCross_Current = xCross.getValue(0);

    if (nCross_Last == null || nCross_Current == null)
        return;

    var bLong = false;
    var bShort = false;

    if (nCross_Current > 0 && nCross_Last < 1)
        bLong = true;
    if (nCross_Current < 1 && nCross_Last > 0)
        bShort = true;

    var nFillPrice = xClose.getValue(0);

    if (getCurrentBarIndex() != 0){

        if (bLong){

            Strategy.doLong("Enter Long", Strategy.CLOSE, Strategy.THISBAR,
Strategy.DEFAULT);
            drawShapeRelative(0, BelowBar1, Shape.UPTRIANGLE, null, fpLongColor,
Text.PRESET, getCurrentBarIndex() + "Entry");
            drawTextRelative(0, BelowBar2, "Long", fpLongColor, null,
Text.PRESET|Text.CENTER|Text.BOLD, null, null, getCurrentBarIndex() + "Entry");
            drawTextRelative(0, BelowBar3, Strategy.getDefaultLotSize() + " @ " +
formatPriceNumber(nFillPrice), fpLongColor, null, Text.PRESET|Text.CENTER|Text.BOLD,
null, null, getCurrentBarIndex() + "EntrySettings");
        };
        if (bShort){

            Strategy.doShort("Enter Short", Strategy.CLOSE, Strategy.THISBAR,
Strategy.DEFAULT);
            drawShapeRelative(0, AboveBar1, Shape.DOWNTRIANGLE, null, fpShortColor,
Text.PRESET, getCurrentBarIndex() + "Exit");
            drawTextRelative(0, AboveBar2, "Short", fpShortColor, null,
Text.PRESET|Text.CENTER|Text.BOLD, null, null, getCurrentBarIndex() + "Exit");
            drawTextRelative(0, AboveBar3, Strategy.getDefaultLotSize() + " @ " +
formatPriceNumber(nFillPrice), fpShortColor, null, Text.PRESET|Text.CENTER|Text.BOLD,
null, null, getCurrentBarIndex() + "ExitSettings");
        };

    };

    return [xAVGTyp, xAVGHaC];
}

function Calc_HaOpen(xOHLC4){

    var nOHLC4 = xOHLC4.getValue(-1);

    if (nOHLC4 == null)
        return null;

    var nHaOpen_Last = ref(-1);

    var nReturnValue = (nOHLC4 + nHaOpen_Last) / 2;

    return nReturnValue;
}

var xHigh = null;
var xLow = null;

function Calc_HaC(xOHLC4, xHaOpen){

    if (getBarState() == BARSTATE_ALLBARS)
    {
        xHigh = high();
        xLow = low();
    }

    var nOHLC4 = xOHLC4.getValue(0);
    var nHaOpen = xHaOpen.getValue(0);
    var nHigh = xHigh.getValue(0);
    var nLow = xLow.getValue(0);

    if (nOHLC4 == null || nHaOpen == null)
        return;

    var nReturnValue = (nOHLC4 + nHaOpen + Math.max(nHigh, nHaOpen) + Math.min(nLow,
nHaOpen)) / 4;

    return nReturnValue;
}

function Calc_Cross(xAVGTyp, xAVGHaC, xClose, xOpen){
    var nClose = xClose.getValue(0);
    var nOpen = xOpen.getValue(0);
    var nAVGTyp = xAVGTyp.getValue(0);
    var nAVGHaC = xAVGHaC.getValue(0);

    var nCross = null;
    var nCross_Last = ref(-1);

    if (nAVGTyp == null || nAVGHaC == null)
        return;

    if (nAVGTyp > nAVGHaC && nClose > nOpen)
        nCross = 1
    else if (nAVGTyp < nAVGHaC && nClose < nOpen)
        nCross = 0;
    else
        nCross = nCross_Last;

    return nCross;
}

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: OCTOBER 2014

In part 3 of his series on exploring charting techniques, which appeared in the September 2014 issue of STOCKS & COMMODITIES ("Creating A Trading Strategy"), author Sylvain Vervoort takes an in-depth look at defining, testing, and using a trading strategy. At thinkorswim, we have used our proprietary scripting language *thinkScript* to build a strategy for detecting trends using this method.

We have made the loading process extremely easy: Simply click on the link http://tos.mx/Hp8IeR and choose *Backtest in thinkorswim*, then choose to rename your study to "SyVerPart3." You can adjust the parameters of these within the *edit studies* window to fine-tune your variables. In the article, Vervoort bases his strategy on a renko chart type. In thinkorswim charts, renko bars can be found under *Style → Range for aggregation type*. Then you can adjust the *range type* to "renko" under the *style* menu as well. A sample chart is shown in Figure 3.

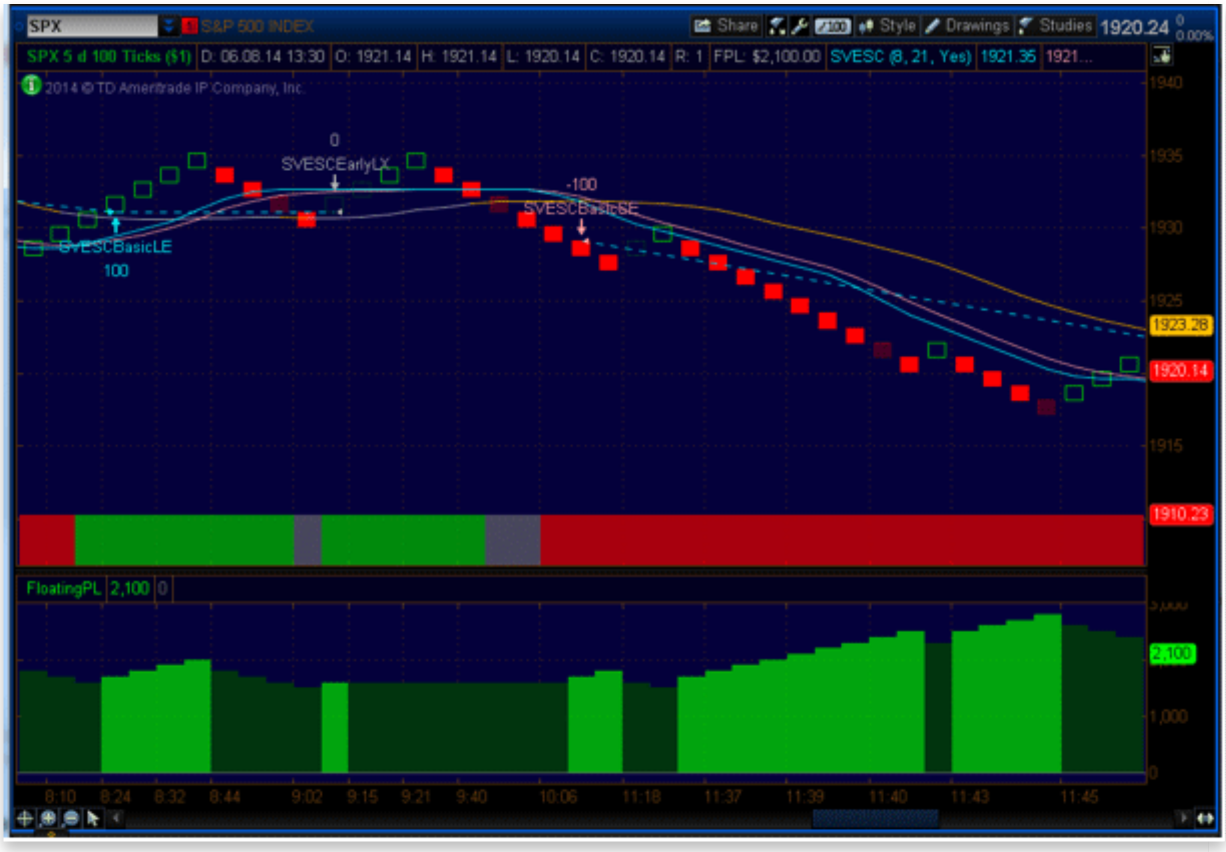


FIGURE 3: THINKORSWIM. Here's an example of the SyVerPart3 study on the SPX.

For a detailed description of the strategy itself, see Vervoort's article in the September 2014 issue of S&C. Happy swimming!

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

BACK TO LIST

WEALTH-LAB: OCTOBER 2014

At first, it may seem that the idea presented in “Exploring Charting Techniques: Creating A Trading Strategy, Part 3” by Sylvain Vervoort, which appeared last month in the September 2014 issue of S&C, is trivial. After all, what don’t we know about the many variations of a moving average crossover? However, in his article, Vervoort takes the technique a step further, applying moving average crossovers on his modified *renko* chart with the added twist of heikin-ashi. The premise is to reduce the noise of a typical fixed-time-related chart and to produce fewer losing trades.

First, we’ll build two moving averages: The fast is the simple moving average (SMA) of the renko-based typical price, and the slow is an SMA of heikin-ashi (HA) recalculated prices. For simplicity of our example strategy, we take a standard renko chart and use daily prices. Despite using the same period, the HA-based average always lags due to added smoothing. The rules of the strategy are:

1. When the eight-period “fast” average crosses above the “slower” counterpart of the same period, a long position is established.
2. When the eight-period “fast” average crosses below the “slower” average of the same period, the long position is closed.

In Figure 4, the green and red renko bricks are superimposed on the open/high/low/close (OHLC) chart.

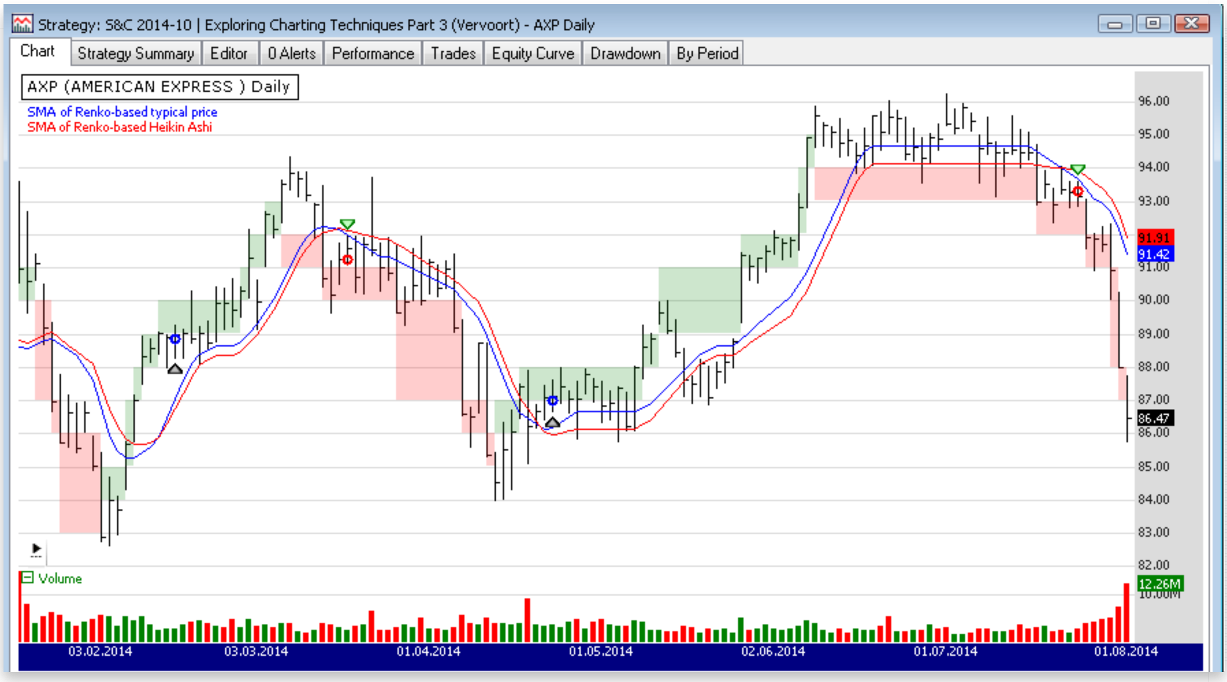


FIGURE 4: WEALTH-LAB. This sample Wealth-Lab 6 chart illustrates application of the system’s rules on a daily chart of AXP (American Express).

To execute the trading system we're providing, Wealth-Lab users can copy & paste the strategy's C# code, or simply let Wealth-Lab do the job: in the *open strategy* dialog, simply click *download* to get the strategy code.

```
C# Strategy code:

using System;
using System.Collections.Generic;
using System.Text;
using System.Drawing;
using WealthLab;
using WealthLab.Indicators;
using WealthLab.ChartStyles.Trending;

namespace WealthLab.Strategies
{
        public class VervoortOct2014 : WealthScript
        {
                StrategyParameter paramRPU;
                StrategyParameter paramPeriod;

                public VervoortOct2014()
                {
                        paramRPU = CreateParameter("Renko Units", 1, 1, 10, 1);
                        paramPeriod = CreateParameter("MA Period", 8, 2, 20, 1);
                }

                protected override void Execute()
                {
                        double rpu = paramRPU.Value;
                        int period = paramPeriod.ValueInt;
                        TRenko renko = new TRenko(Bars, rpu);
                        DataSeries dsOpen = new DataSeries(Bars,"R-Open"), dsHigh =
new DataSeries(Bars,"R-High"),
                                dsLow = new DataSeries(Bars,"R-Low"), dsClose = new
DataSeries(Bars,"R-Close");
                        DataSeries haOpen = new DataSeries(Bars,"HA-Open"), haHigh =
new DataSeries(Bars,"HA-High"),
                                haLow = new DataSeries(Bars,"HA-Low"), haClose = new
DataSeries(Bars,"HA-Close");

                        for(int bar = 1; bar < Bars.Count; bar++)
                        {
                                Renko rko = renko.Columns[bar];

                                // Create Renko-based OHLC and Heikin Ashi for
averaging
                                if( rko.Col > -1 )
                                {
                                        if( rko.Col > renko.Columns[bar-1].Col )
                                        {
                                                Renko prev = renko.Columns[bar-1];

                                                double open = rko.DirectionUp ?
rko.Low : rko.High;
                                                double close = !rko.DirectionUp ?
rko.Low : rko.High;
                                                double high = rko.High;
                                                double low = rko.Low;

                                                double prevOpen = prev.DirectionUp ?
prev.Low : prev.High;
                                                double prevClose = !prev.DirectionUp
? prev.Low : prev.High;
                                                double prevHigh = prev.High;
                                                double prevLow = prev.Low;

                                                double _haClose = (open + high + low
+ close) / 4;
                                                double _haOpen = (prevOpen +
prevClose) / 2;
                                                double _haHigh = Math.Max(
Math.Max(high, open), close );
                                                double _haLow = Math.Min(
Math.Min(low, open), close );

                                                dsOpen[bar] = open;
                                                dsHigh[bar] = high;
                                                dsLow[bar] = low;
                                                dsClose[bar] = close;

                                                haOpen[bar] = _haOpen;
                                                haHigh[bar] = _haHigh;
                                                haLow[bar] = _haLow;
                                                haClose[bar] = _haClose;
                                        }
                                        else
                                        {
                                                dsOpen[bar] = dsOpen[bar-1];
                                                dsHigh[bar] = dsHigh[bar-1];
                                                dsLow[bar] = dsLow[bar-1];
                                                dsClose[bar] = dsClose[bar-1];

                                                haOpen[bar] = haOpen[bar-1];
                                                haHigh[bar] = haHigh[bar-1];
                                                haLow[bar] = haLow[bar-1];
                                                haClose[bar] = haClose[bar-1];
                                        }
                                }
                        }

                        //The first and faster average is the SMA of the typical
price (HLC/3)
                        //The second average is a SMA of heikin ashi re-calculated
prices or haOpen + haHigh + haLow + haClose divided by four

                        DataSeries maTypical = (SMA.Series( dsHigh, period ) +
SMA.Series( dsLow, period ) + SMA.Series( dsClose, period )) / 3;
                        DataSeries maHeikin = (SMA.Series( haOpen, period ) +
SMA.Series( haHigh, period ) + SMA.Series( haLow, period ) + SMA.Series( haClose,
period )) / 4;

                        maTypical.Description = "SMA of Renko-based typical price";
                        maHeikin.Description = "SMA of Renko-based Heikin Ashi";

                        PlotSeries( PricePane, maTypical, Color.Blue,
LineStyle.Solid, 1 );
                        PlotSeries( PricePane, maHeikin, Color.Red, LineStyle.Solid,
1 );


                        for(int bar = 1; bar < Bars.Count; bar++)
                        {
                                // Detect crossover/crossunder and store state in a
variable
                                bool maXo = CrossOver(bar, maTypical, maHeikin);
                                bool maXu = CrossUnder(bar, maTypical, maHeikin);

                                Position p =
LastPosition;
                                if ( IsLastPositionActive )
                                {
                                        if ( maXu )
                                                SellAtMarket( bar + 1, p );
                                }
                                else
                                {
                                         if ( maXo )
                                                BuyAtMarket( bar + 1 );
                                }

                        }

                }
        }
}
```

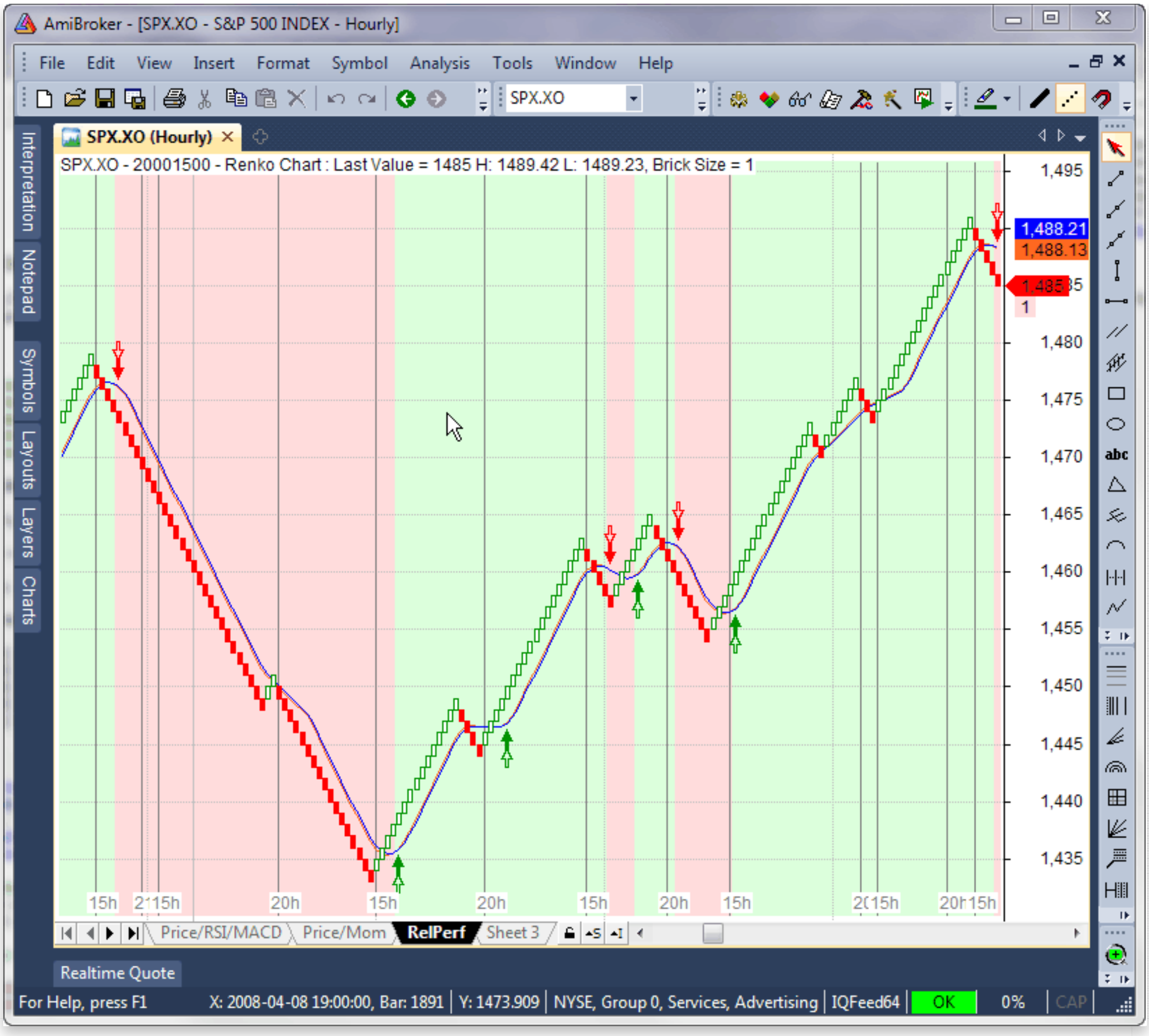
—Eugene, Wealth-Lab team
MS123, LLC
www.wealth-lab.com

BACK TO LIST

AMIBROKER: OCTOBER 2014

In “Exploring Charting Techniques: Creating A Trading Strategy, Part 3,” which appeared in the September 2014 issue of STOCKS & COMMODITIES, author Sylvain Vervoort continued his article series presenting a trading system based on modified renko charts and moving averages.

We are providing a ready-to-use formula for AmiBroker. It is based on the formula that Vervoort presented in his September 2014 article with the addition of trading rules, a colorized background, and moving averages for display in AmiBroker (see Figure 5).



SPX.XO - 20001500 - Renko Chart : Last Value = 1485 H: 1489.42 L: 1489.23, Brick Size = 1

For Help, press F1 | X: 2008-04-08 19:00:00, Bar: 1891 | Y: 1473.909 | NYSE, Group 0, Services, Advertising | IQFeed64 | OK | 0% | CAP

FIGURE 5: AMIBROKER. This modified renko chart of the S&P 500 index displays moving average crossovers and sample trading system entry/exit points.

```
// Modified Renko Chart with custom date axis
// and high/low winks
// Loosely based on Renko chart formula by G. Kavanagh
// from AmiBroker on-line formula library (id=521)
// Modifications & fixes TJ 2014

function FillRun( dir, num, changedir )
{
       global i, j, modified, dt, RKC, RKO, RKD, RKH, RKL;

       for ( x = 1; x <= num AND j < BarCount - 1; x++ )
       {
           j++;

           extra = ( changedir AND x == 1 ) * dir;

           RKC[ j ] = RKC[ j - 1 ] + dir + extra;
           RKO[ j ] = RKC[ j - 1 ] + IIf( modified, 0, extra );
           RKD[ j ] = dt[ i ];
           RKH[ j ] = High[ i - 1 ];
           RKL[ j ] = Low[ i - 1 ];
       }
}

SetBarsRequired( sbrAll, sbrAll );
Brick = Param( "Brick Size", 0.001, 0.0001, 1.00, 0.001 );
reverse = 2;
intra = ParamToggle( "Intraday", "No|Yes", 0 );
modified = ParamToggle( "Modified", "No|Yes", 0 );

// Convert the closing price to rising and falling rounded bricks
CF = ceil( C / Brick );
CR = floor( C / Brick );

// initialize first element
j = 0;
RKC[j] = CF[0];
RKO[j] = CF[0] + 1;
RKD = 0;
RKH = 0;
RKL = 0;
dt = IIf( intra, floor( TimeNum() / 100 ), DateNum() );

dir = -1; // 1 is up, -1 is down

// Loop to produce the Renko values in number of bricks

for ( i = 1; i < BarCount - 1; i++ )
{
   if ( j >= BarCount )
       break; // no more room -> finish

   if ( CF[i] <= RKC[j] - 1 AND dir < 0 ) // Continue down
   {
       num = RKC[j] - CF[i];

       FillRun( dir, num, False );
   }
   else
   if ( CR[i] >= RKC[j] + Reverse AND dir < 0 ) // Change down to up
   {
       num = CR[i] - RKC[j];
       dir = 1;

       FillRun( dir, num, True );
   }
   else
   if ( CR[i] >= RKC[j] + 1 AND dir > 0 ) // Continue Up
   {
       num = CR[i] - RKC[j];
       FillRun( dir, num, False );
   }
   else
   if ( CF[i] <= RKC[j] - Reverse AND dir > 0 ) // Change up to down
   {
       num = RKC[j] - CF[i];
       dir = -1;
       FillRun( dir, num, True );
    }
}

// move the chart to right end of chart space, ie last brick on last bar position
delta =  BarCount - 1 - j;

RKC = Ref( RKC, -delta );
RKO = Ref( RKO, -delta );
RKD = Ref( RKD, -delta );
RKH = Ref( RKH, -delta );
RKL = Ref( RKL, -delta );

C = RKC * Brick;
O = RKO * Brick;
H = IIf( modified, RKH, Max( C, O ) );
L = IIf( modified, RKL, Min( C, O ) );

Plot( C, "", IIf( C > O, colorGreen, colorRed ), styleCandle );

m1 = MA( (C+H+L)/3, 8 );
m2 = MA( C, 8 );

Plot( m1, "SMA Typ", colorBlue );
Plot( m2, "SMA Renko", colorOrange );

Cover = Cross( m2, m1 );
Sell = Cross( m1, m2 );

Short = Sell AND C < O;
Buy = Cover AND C > O;

PlotShapes( shapeUpArrow * Buy, colorGreen, 0, m1);
PlotShapes( shapeDownArrow * Sell, colorRed, 0, m1 );
PlotShapes( shapeHollowUpArrow * Cover, colorGreen, 0, m1, -25);
PlotShapes( shapeHollowDownArrow * Short, colorRed, 0, m1, -25 );

color = IIf( Flip( Buy, Sell ), ColorRGB( 220, 255, 220 ),
         IIf( Flip( Short, Cover ), ColorRGB( 255, 220, 220 ), colorWhite ) );

Plot( 1, "", color, styleArea | styleOwnScale, 0, 1, 0, -1 );

xnum = floor( RKD / 1000 );
XChange = IIf( xnum != Ref( xnum, -1 ), 1, Null );

Plot( XChange, "", colorGrey50, styleHistogram | styleOwnScale, 0, 1 );

// Draw renko-date axis
MonthNames = "Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec";
fvb = FirstVisibleValue( BarIndex() );
lvb = LastVisibleValue( BarIndex() );

for ( i = fvb; i < lvb; i++ )
{
 if ( XChange[ i ] )
 {
   if ( intra )
      datetext = StrFormat( "%02gh", floor ( RKD[ i ] / 100 ) );
   else
```

```
    if ( ( xnum[ i ] % 100 ) == 1 )
       datetext = StrFormat( "%04.0f", 1900 + ( xnum[ i ] / 100 ) );
    else
       datetext = StrExtract( MonthNames, ( xnum[ i ] % 100 ) - 1 );

    PlotText( datetext  , i, LowestVisibleValue( Low ), colorGrey50, colorWhite, -20
);
  }
}

Title = Name() + StrFormat( " - 20%06.0f", RKD % 1000000 ) + " - Renko Chart : Last
Value = " + RKC * Brick + " H: " + RKH + " L: " + RKL + ", Brick Size = " + Brick;

GraphXSpace = 5;
```

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

BACK TO LIST

NEUROSHELL TRADER: OCTOBER 2014

We have recreated the renko bar trading system described by Sylvain Vervoort in his September 2014 article "Exploring Charting Techniques: Creating A Trading Strategy, Part 3," using NeuroShell Trader's point-and-click indicator wizard without the need for programming.

We used the InterChart Tools Renko add-in to NeuroShell and the heikin-ashi close indicator from a previous Traders' Tip to quickly set up the eight-period simple moving averages of the *typical price* and the *heikin-ashi average closing price* (see Vervoort's article in the September 2014 issue for more details of his technique).

To produce a chart similar to the one we show in Figure 6 of the S&P 500 index, you can insert the indicators as follows:

1. Select "New indicator" from the Insert menu
2. Choose the *averages* category and select *simple moving average, 8 periods*
3. Substitute the Ict Renko HLC3 (0.10, 1, 1, 10, High, Low, Volume) for the default value to recreate the *typical price* indicator described in Vervoort's article
4. Create another average indicator, and this time, substitute the HeikinAshiClose of the corresponding Ict Renko Bars to generate the *heikin-ashi average closing price* indicator.

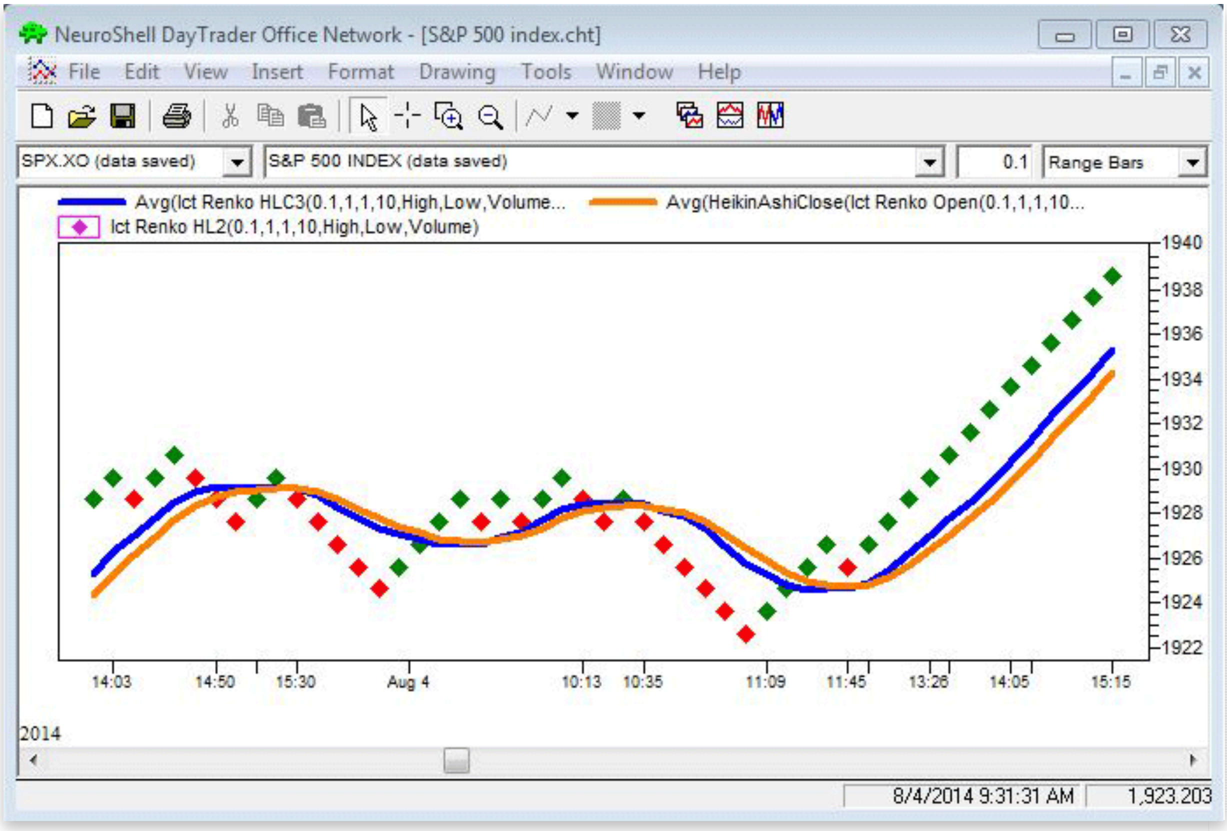

NeuroShell DayTrader Office Network - [S&P 500 index.cht]
File Edit View Insert Format Drawing Tools Window Help
SPX.XO (data saved)
S&P 500 INDEX (data saved)
0.1
Range Bars
Avg(Ict Renko HLC3(0.1,1,1,10,High,Low,Volume...
Avg(HeikinAshiClose(Ict Renko Open(0.1,1,1,10...
Ict Renko HL2(0.1,1,1,10,High,Low,Volume)
1940
1938
1936
1934
1932
1930
1928
1926
1924
1922
14:03
14:50
15:30
Aug 4
10:13
10:35
11:09
11:45
13:26
14:05
15:15
2014
8/4/2014 9:31:31 AM
1,923.203

FIGURE 6: NEUROSHELL TRADER. This NeuroShell Trader chart displays the crossover of the eight-period SMA of the typical price and heikin-ashi average closing price. This strategy was created using the indicator wizard in NeuroShell Trader, so no programming is needed by the user.

The InterChart Tools Renko bars are virtual bars and perform their calculations using the same methods as traditional renko bars, but once a trading signal is generated by the renko bar, both the trade and fill are correctly displayed on the open of the next bar of the base chart.

The base chart is a 0.10 range bar of the S&P 500 index. The value of 0.10 virtual tick size in the Ict Renko Bars corresponds to the size of the base chart's range bar. The next two parameters represent the number of ticks used to calculate the up part of the renko bar, followed by the number of ticks used to compute the down part. The "10" represents a multiplier that is applied to the described renko bar's up/down ratio to realize its final size. This enables the indicators to use a different number of ticks for the up and down side of the renko bars. Since any bar's function is to absorb noise, and rising price jitter is often different from falling price jitter, our renko bars permit an asymmetrical definition to accommodate this.

In the trading system described by Vervoort in his article, the trading signals occur when the average of the typical price crosses above or below the average of the heikin-ashi close of the renko bars. Rather than using a visual system, you could use NeuroShell Trader's point-and-click wizard to build the crossover trading rules and allow NeuroShell Trader's optimizer to identify the optimal bar size and noise absorption for a given algorithm or equity.

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

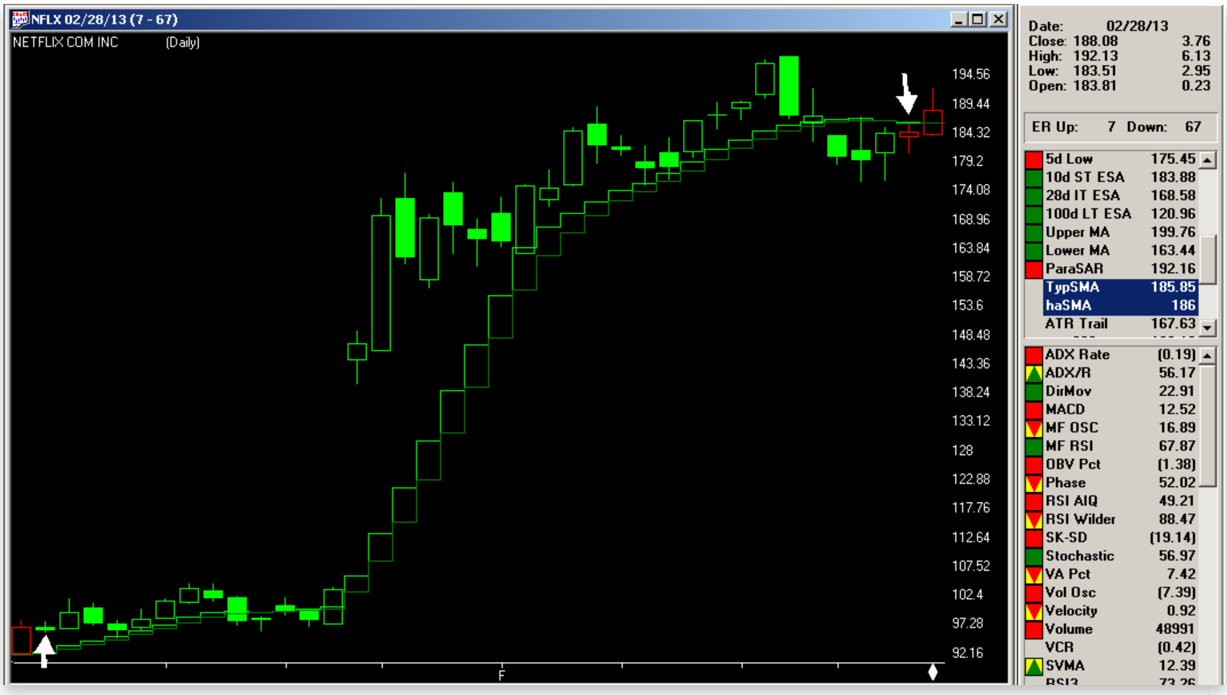
AIQ: OCTOBER 2014

The AIQ code for this month is based on Sylvain Vervoort's article in the September 2014 issue of STOCKS & COMMODITIES, "Exploring Charting Techniques: Creating A Trading Strategy, Part 3."

The code and EDS file can be downloaded from www.TradersEdgeSystems.com/traderstips.htm, and is also shown below.

I was not able to code the renko-based charts, so the two moving averages, the simple average of the typical price (typSMA), and the simple average of the heikin-ashi close (haSMA) are based on the closing values from a conventional chart, not the renko chart.

In Figure 7, I show a chart of Netflix (NFLX) with the typSMA and haSMA averages. The chart also shows a trade that was opened on 1/4/2013 and closed on 2/27/2013 for a 90% profit.



Date:	02/28/13	
Close:	188.08	3.76
High:	192.13	6.13
Low:	183.51	2.95
Open:	183.81	0.23

ER Up: 7 Down: 67

🟥	5d Low	175.45
🟩	10d ST ESA	183.88
🟩	28d IT ESA	168.58
🟩	100d LT ESA	120.96
🟩	Upper MA	199.76
🟩	Lower MA	163.44
🟥	ParaSAR	192.16
	TypSMA	185.85
	haSMA	186
	ATR Trail	167.63

🟥	ADX Rate	(0.19)
▲	ADX/R	56.17
🟩	DirMov	22.91
🟥	MACD	12.52
▼	MF OSC	16.89
🟩	MF RSI	67.87
🟥	OBV Pct	(1.38)
▼	Phase	52.02
🟥	RSI AIQ	49.21
▼	RSI Wilder	88.47
🟥	SK-SD	(19.14)
🟩	Stochastic	56.97
▼	VA Pct	7.42
🟥	Vol Osc	(7.39)
▼	Velocity	0.92
🟥	Volume	48991
	VCR	(0.42)
▲	SVMA	12.39
	RSI3	73.26

FIGURE 7: AIQ. Here is a sample chart of Netflix (NFLX) with the typSMA and haSMA averages plus a sample trade marked with white up and down arrows.

```
!Exploring Charting Techniques: Creating A Trading Strategy
!Author: Sylvain Vervoort, TASC Oct 2014
!Coded by: Richard Denning 08/03/2014
!www.TradersEdgeSystems.com

!NOTE: Renko based charts are not available in AIQ. The following
!      code is conventional open, high, low , close charts.

! INPUTS:
H is [high].
L is [low].
C is [close].
O is [open].
OSD is offsettodate(month(),day(),year()).
typLen is 8.
haLen is 8.

!----------------HEIKIN-ASHI-----------------
haC is (O + H +L + C) / 4.
DaysInto is ReportDate() - RuleDate().
end if DaysInto > 20.
endHAO is iff(end,O, haO).
haO is (valresult(endHAO,1)
        +valresult(haC,1))/2.
haH is Max(H,max(haO,haC)).
haL is Min(L,min(haO,haC)).
haCL is (haC + haO + haH + haL) / 4.
haSMA is simpleavg(haCL,haLen).  !PLOT
!----------------end HEIKIN-ASHI---------------
```

```
!----------------TYPICAL PRICE ----------------
TYP is (H+L+C)/3.
typSMA is simpleavg(TYP,typLen). !PLOT

!-----------------end TYPICAL-----------------

!--------------COLOR STUDY--------------------
GREEN if typSMA > haSMA.
RED if typSMA < haSMA.
!-------------end COLOR STUDY-----------------

!-------------TRADING SYSTEM------------------
Buy if GREEN.
ExitBuy if RED.
Sell if RED.
ExitSell if GREEN.
!-------------end TRADING SYSTEM--------------
```

—Richard Denning
info@TradersEdgeSystems.com
for AIQ Systems

BACK TO LIST

TRADERSSTUDIO: OCTOBER 2014

The TradersStudio code I am providing for Sylvain Vervoort's September 2014 article in S&C, "Exploring Charting Techniques: Creating A Trading Strategy, Part 3," can be found at the following two websites:

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

The following code files are provided in the download:

- Function HA_SMA: Returns the heikin-ashi simple moving average of the close based on the "haLen" input
- Function TYP_SMA: Returns the typical-price simple moving average of the close based on the typLen input
- Indicator plot TYP_HA_SMA: For plotting the two moving averages just described on a chart
- System SVE_TYP_HA_SMA_SYS: The system code for trading crossovers on the two moving averages.

I was not able to code the renko-based charts, so the two moving averages, the simple average of the typical price (typSMA), and the simple average of the heikin-ashi close (haSMA) are based on the closing values from a conventional chart, not the renko chart.

In Figure 8, I show a chart of the S&P 500 full-size futures contract (SP) using data from Pinnacle Data Corp. (www.pinnacledata.com) with the typSMA and haSMA averages. In addition, this chart shows a sample trade that was opened on 10/14/2013 and closed on 11/11/2013 for a $19,875 profit before commission & slippage.

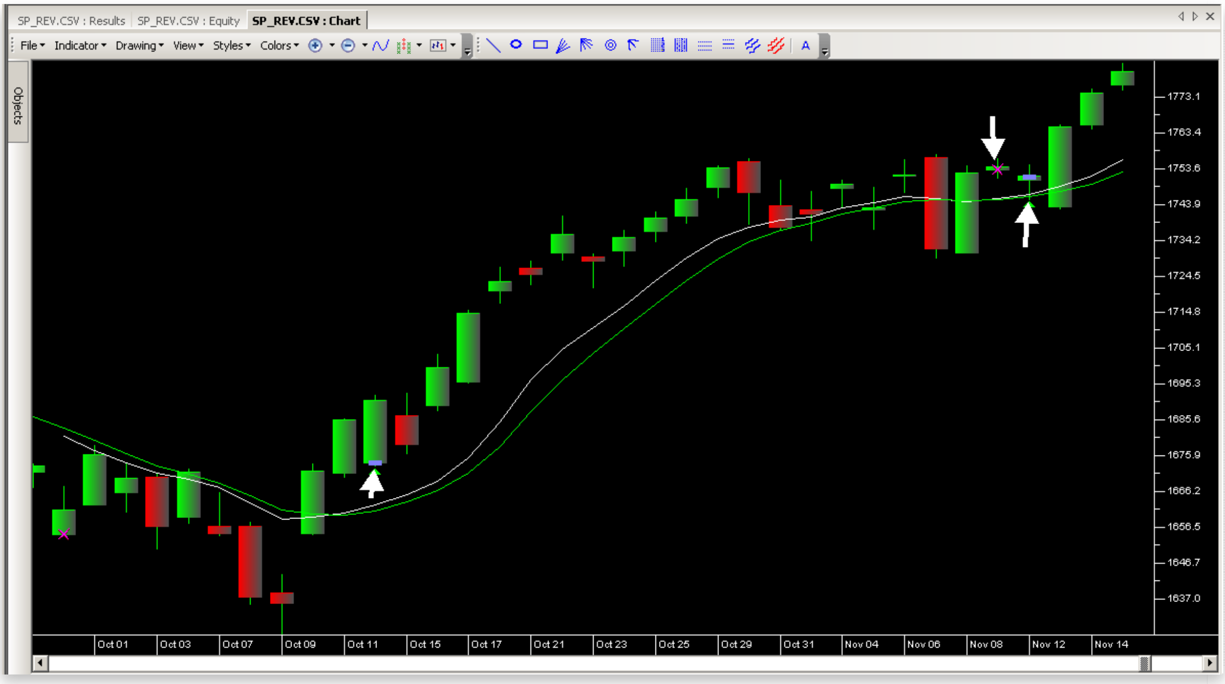


FIGURE 8: TRADERSSTUDIO. Here is an example crossover of the typSMA and haSMA averages on a chart of the S&P 500 full-size futures contract (SP). The trade shown here, which was opened on 10/14/2013 and closed on 11/11/2013, would have produced a profit of $19,875 before commission & slippage.

```
'Exploring Charting Techniques: Creating A Trading Strategy
'Author: Sylvain Vervoort, TASC Oct 2014
'Coded by: Richard Denning 08/03/2014
'www.TradersEdgeSystems.com

'NOTE: Renko based charts are not available in AIQ. The following
'      code is conventional open, high, low , close charts.

Function HA_SMA(haLen)
    Dim haC As BarArray
    Dim haO As BarArray
    Dim haH As BarArray
    Dim haL As BarArray
    Dim haCL As BarArray
    haC = (O+H+L+C)/4
    haO = IIF(BarNumber = FirstBar,O,(haO[1] + haC[1])/2)
    haH = Max(H,haO)
    haL = Min(L,haO)
    haCL = (haC+haO+haH+haL)/4
```

```
HA_SMA = Average(haCL,haLen)
End Function
'------------------------------------------------------------------
Function TYP_SMA(typLen)
Dim TYP As BarArray
TYP = (H+L+C)/3
TYP_SMA = Average(TYP,typLen)
End Function
'------------------------------------------------------------------
Sub TYP_HA_SMA(typLen,haLen)
plot1(typ_SMA(typLen))
plot2(ha_SMA(haLen))
End Sub
'------------------------------------------------------------------
Sub SVE_TYP_HA_SMA_SYS(typLen,haLen,LSB)
    'typLen = 8,haLen = 8
    'LSB = 1 (Longs only)
       ' = -1 (Shorts only)
       ' = 0 (Both Longs and Shorts)
    Dim typSMA As BarArray
    Dim haSMA As BarArray
    Dim Green, Red
    typSMA = TYP_SMA(typLen)
    haSMA = HA_SMA(haLen)
    Green = typSMA > haSMA And C > O
    Red = typSMA < haSMA And C > O
    If Green And (LSB = 1 Or LSB = 0) Then Buy("LE",1,0,Market,Day)
    If Red Then ExitLong("LX","",1,0,Market,Day)
    If Red And (LSB = 0 Or LSB = -1) Then Sell("SE",1,0,Market,Day)
    If Green Then ExitShort("SX","",1,0,Market,Day)
End Sub
'-------------------------------------------------------------------
```

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

NINJATRADER: OCTOBER 2014

The SveRenkoCross strategy, as introduced by Sylvain Vervoort in the September 2014 STOCKS & COMMODITIES article "Exploring Charting Techniques: Creating A Trading Strategy, Part 3," has been made available for download at www.ninjatrader.com/SC/October2014SC.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 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 "SveRenkoCross" file. A sample chart implementing the strategy is shown in Figure 9.

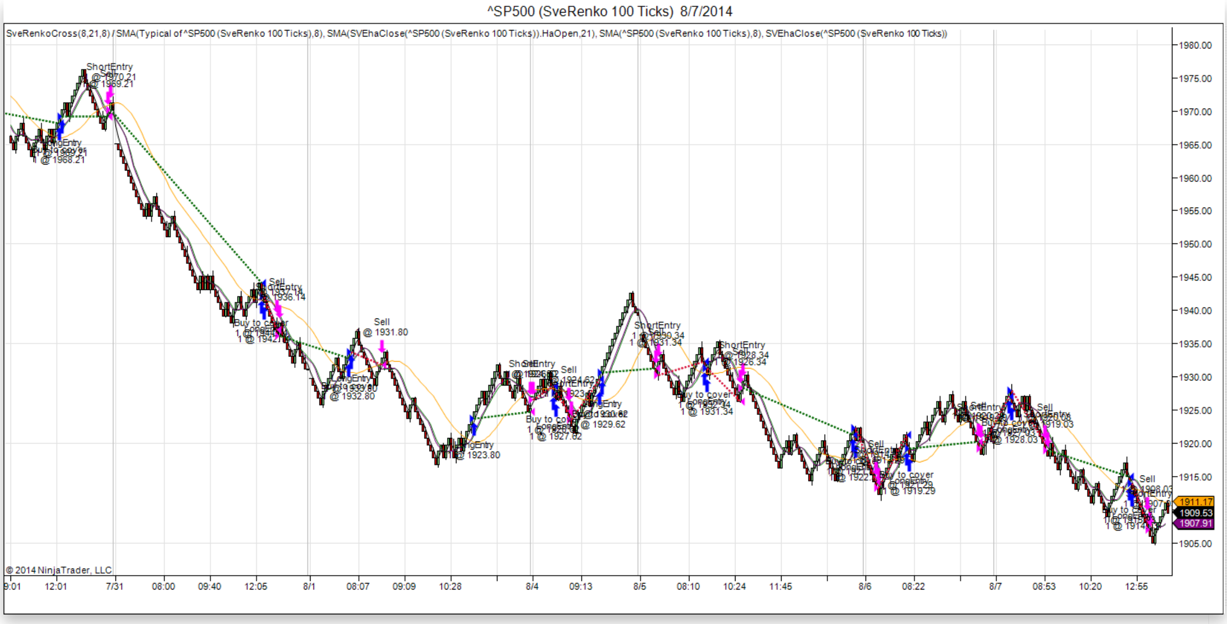


FIGURE 9: NINJATRADER. This screenshot shows the strategy applied to a 100-tick SveRenko S&P 500 chart in NinjaTrader.

—Raymond Deux & Cal Hueber
NinjaTrader, LLC
www.ninjatrader.com

BACK TO LIST

UPDATA: OCTOBER 2014

Our Traders' Tip this month is one that was developed in-house by our Updata team and is for a trading system we named the *small-range bars system*.

This system is a daily breakout system that identifies when the previous day's absolute daily range (high to low) and normalized by the close is at least half a standard deviation below its cumulative period average. The system anticipates the next day will be of larger range. On that day, signals are generated upon breaking of the previous day's high or low. All positions are flattened at the close.

The Updata code for this system 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.

```
'Small Range Bars System
DISPLAYSTYLE 3LINES
INDICATORTYPE CHART
INDICATORTYPE2 TOOL
PLOTSTYLE2 LINE RGB(200,0,0)
NAME "Abs Range With Cumulative Avg. - 0.5 Std.Deviations" ""
@PCTBAR=0
@STDDEVBAR=0
@AVGBAR=0
#Inc=0
@VOLUMEAVG=0
@SUMBAR=0
@BARAVG=0
@DIFFSq=0

FOR #CURDATE=1 TO #LASTDATE
    IF ABS(HIGH-LOW)>0
       #Inc=#Inc+1
       @PCTBAR=(ABS(HIGH(1)-LOW(1)))/CLOSE(1)
       @SUMBAR=@SUMBAR+@PCTBAR
       @BARAVG=@SUMBAR/#Inc
       @DIFFSq=@DIFFSq+((@PCTBAR-@BARAVG)*(@PCTBAR-@BARAVG))
       @STDDEVBAR=EXPBASE(@DIFFSq/(#Inc-1),0.5)
       'ENTRIES IF BAR IS 0.5 STD DEV BELOW AVG RANGE
       'LONGS UPON BREAK OF PREV HIGH
       'SHORTS UPON BREAK OF PREV LOW
       IF HIST(@PCTBAR<(@BARAVG-0.5*@STDDEVBAR),1)
          IF HIGH>HIGH(1)
             BUY MAX(HIGH(1),OPEN)
          ENDIF
          IF LOW<LOW(1)
             SHORT MIN(LOW(1),OPEN)
          ENDIF
       ENDIF
       'CLOSE POSITIONS ON SAME BAR
       SELL CLOSE
       COVER CLOSE
       @PLOT=@PCTBAR
       @PLOT2=@BARAVG-0.5*@STDDEVBAR
    ENDIF
NEXT
```

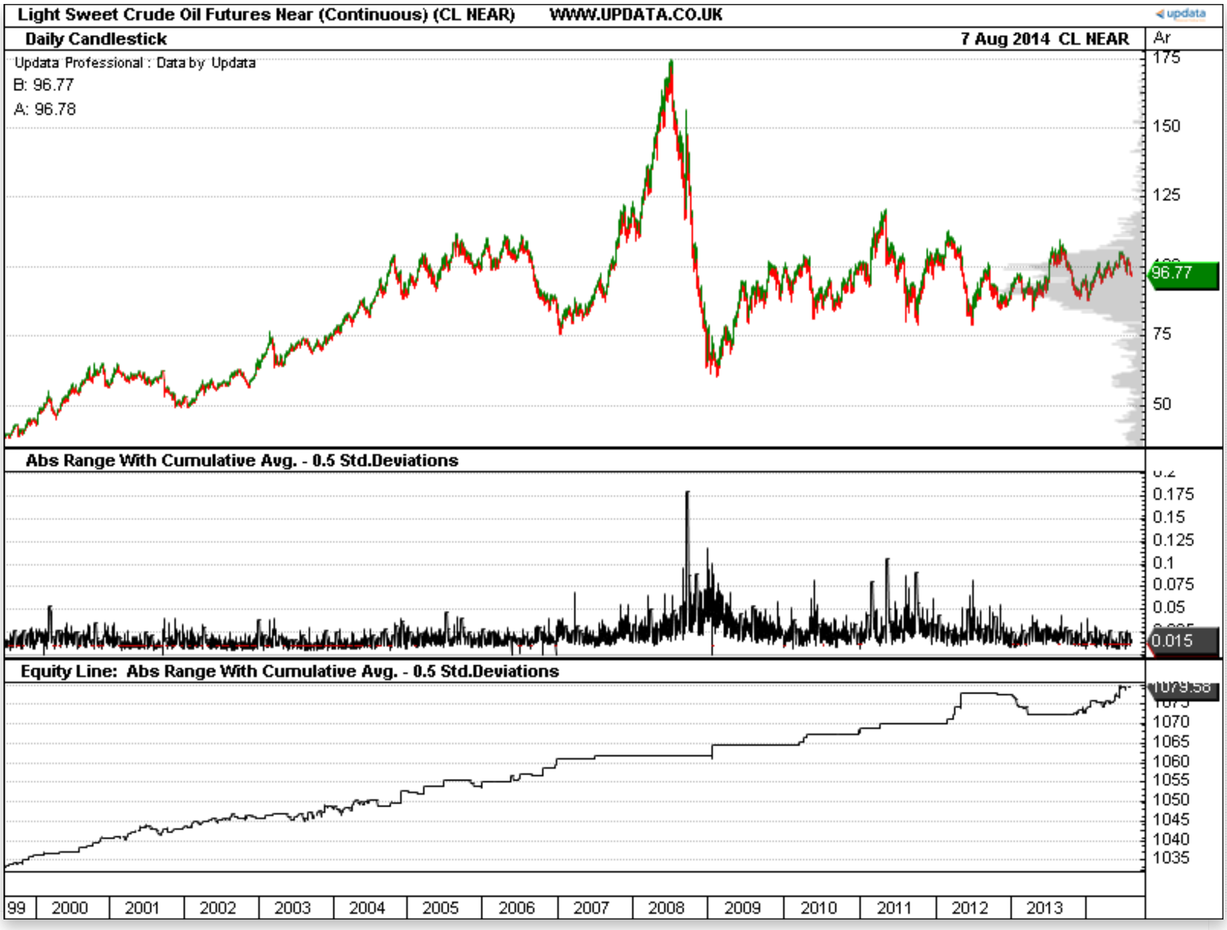


FIGURE 10: UPDATA, SMALL-RANGE BARS SYSTEM. This chart shows an example of our small-range bars system as applied to NYMEX-listed WTI crude oil prices.

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

BACK TO LIST

MICROSOFT EXCEL: OCTOBER 2014

In “Exploring Charting Techniques: Creating A Trading Strategy, Part 3” by Sylvain Vervoort, which appeared in the September 2014 issue of STOCKS & COMMODITIES, the author develops a simple trading strategy around the crossover of two moving averages constructed over the modified renko brick chart that he showed us earlier in his July 2014 S&C article of the same series (part 1).

I played around with the renko tick size (700), which translates to a brick size of $700. With the S&P 500 as the datastream (which is a big ticket compared to Ford at $17 a share), this allowed more dates on the renko chart for demonstration purposes. Adjust as you like for your trading instrument.

I also made one small change to his signal strategy. Rather than use a set 0.8 signal threshold, my spreadsheet calculates the maximum absolute delta between the moving averages and then takes a user-specified percentage of that delta as the threshold value. This allows the strategy to adapt to tradable instruments with large differences in pricing scales.

This strategy seems to work equally well with tick-level data or end-of-day data.

Unless we get a sharp signal crossover (as we see near 12/18/2013 in Figure 11), the delta between the two moving averages can be less than our signal threshold for one or more bars, and we get white space.

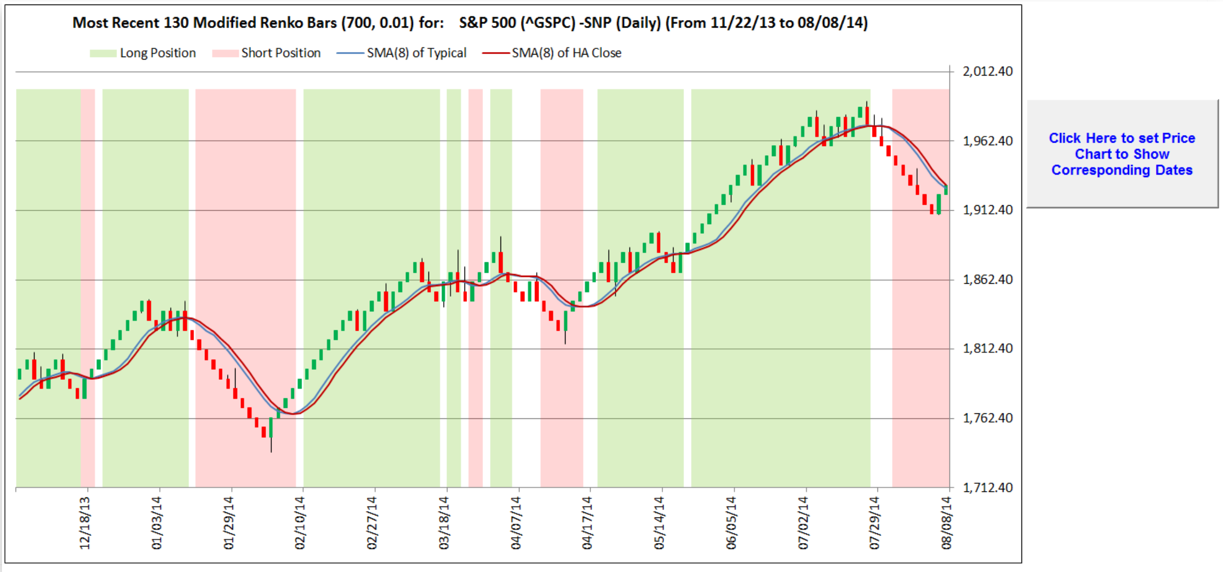
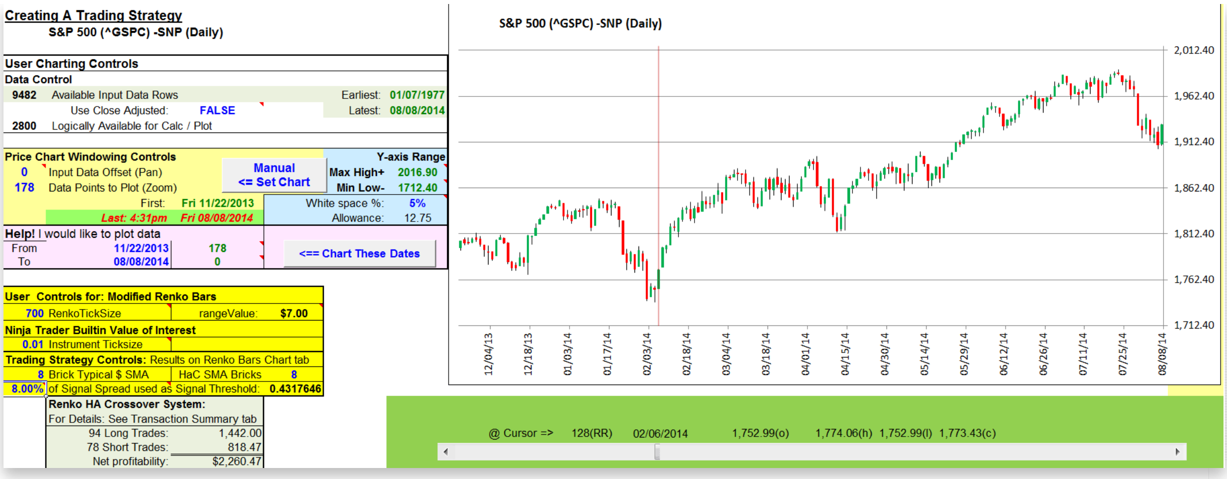


FIGURE 11: EXCEL, CROSSOVER ON RENKO CHART. This shows a renko chart with the position durations per indicator. The crossover spread exceeds the signal threshold.

The chart in Figure 11 has a fixed number of bars to avoid some clutter. The right-most bar is for the same date as the right-most bar on the price chart in Figure 12. Because of the nature of the renko bar construct, the number of renko bars for a given time span will almost never be the same as the number of source data bars for the same time span.

To help get the charts to cover the same time span for the sake of visual comparison, the button to the right of the renko chart will reset the price chart of Figure 12 to the time span shown in the renko chart.



Creating A Trading Strategy

S&P 500 (^GSPC) -SNP (Daily)

User Charting Controls

Data Control

9482	Available Input Data Rows	Earliest:	01/07/1977
	Use Close Adjusted: FALSE	Latest:	08/08/2014
2800	Logically Available for Calc / Plot		

Price Chart Windowing Controls | **Y-axis Range**

0	Input Data Offset (Pan)	Max High+	2016.90
178	Data Points to Plot (Zoom)	Min Low-	1712.40
First:	Fri 11/22/2013	White space %:	5%
Last: 4:31pm	*Fri 08/08/2014*	Allowance:	12.75

Help! I would like to plot data

From	11/22/2013	178
To	08/08/2014	0

User Controls for: Modified Renko Bars

700	RenkoTickSize	rangeValue:	$7.00

Ninja Trader Builtin Value of Interest

0.01	Instrument Ticksize

Trading Strategy Controls: Results on Renko Bars Chart tab

8	Brick Typical $ SMA	HaC SMA Bricks	8
8.00%	of Signal Spread used as Signal Threshold:		0.4317646

Renko HA Crossover System:

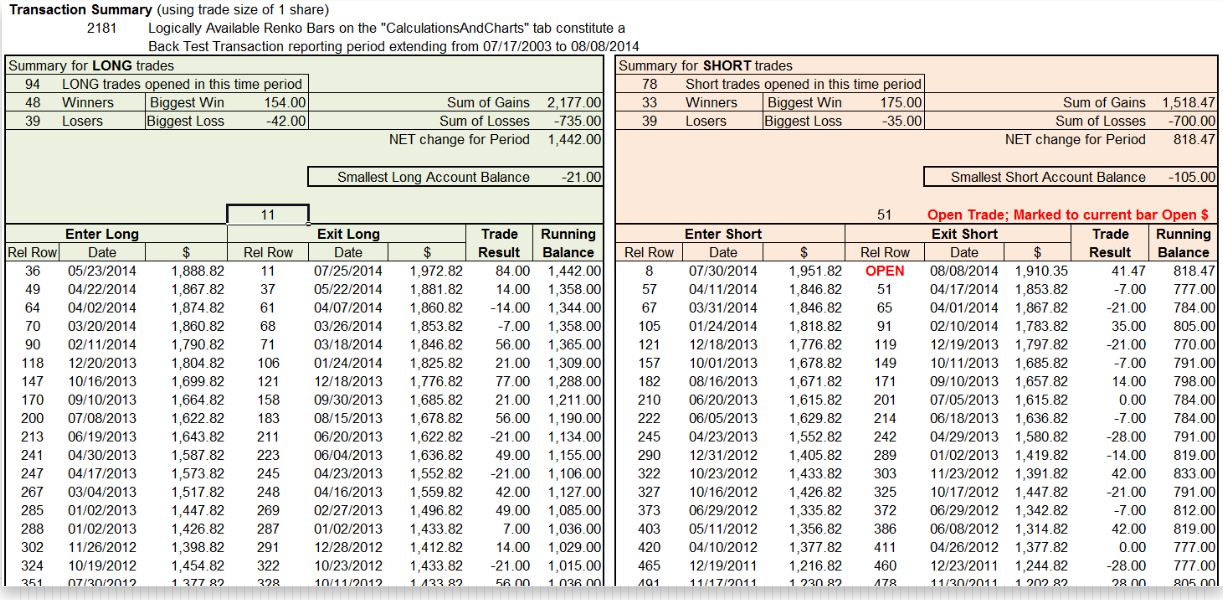
For Details: See Transaction Summary tab

94 Long Trades:	1,442.00
78 Short Trades:	818.47
Net profitability:	$2,260.47

@ Cursor => 128(RR) 02/06/2014 1,752.99(o) 1,774.06(h) 1,752.99(l) 1,773.43(c)

FIGURE 12: EXCEL. Here's a sample price chart for comparison to the renko chart in Figure 11.

Figure 13 details the backtest transactions that were summarized in the blue box at the bottom left of the chart in Figure 12.



Transaction Summary (using trade size of 1 share)

2181 Logically Available Renko Bars on the "CalculationsAndCharts" tab constitute a
Back Test Transaction reporting period extending from 07/17/2003 to 08/08/2014

Summary for **LONG** trades

94	LONG trades opened in this time period				
48	Winners	Biggest Win	154.00	Sum of Gains	2,177.00
39	Losers	Biggest Loss	-42.00	Sum of Losses	-735.00
				NET change for Period	1,442.00
				Smallest Long Account Balance	-21.00

11

Enter Long			Exit Long			Trade	Running
Rel Row	Date	$	Rel Row	Date	$	Result	Balance
36	05/23/2014	1,888.82	11	07/25/2014	1,972.82	84.00	1,442.00
49	04/22/2014	1,867.82	37	05/22/2014	1,881.82	14.00	1,358.00
64	04/02/2014	1,874.82	61	04/07/2014	1,860.82	-14.00	1,344.00
70	03/20/2014	1,860.82	68	03/26/2014	1,853.82	-7.00	1,358.00
90	02/11/2014	1,790.82	71	03/18/2014	1,846.82	56.00	1,365.00
118	12/20/2013	1,804.82	106	01/24/2014	1,825.82	21.00	1,309.00
147	10/16/2013	1,699.82	121	12/18/2013	1,776.82	77.00	1,288.00
170	09/10/2013	1,664.82	158	09/30/2013	1,685.82	21.00	1,211.00
200	07/08/2013	1,622.82	183	08/15/2013	1,678.82	56.00	1,190.00
213	06/19/2013	1,643.82	211	06/20/2013	1,622.82	-21.00	1,134.00
241	04/30/2013	1,587.82	223	06/04/2013	1,636.82	49.00	1,155.00
247	04/17/2013	1,573.82	245	04/23/2013	1,552.82	-21.00	1,106.00
267	03/04/2013	1,517.82	248	04/16/2013	1,559.82	42.00	1,127.00
285	01/02/2013	1,447.82	269	02/27/2013	1,496.82	49.00	1,085.00
288	01/02/2013	1,426.82	287	01/02/2013	1,433.82	7.00	1,036.00
302	11/26/2012	1,398.82	291	12/28/2012	1,412.82	14.00	1,029.00
324	10/19/2012	1,454.82	322	10/23/2012	1,433.82	-21.00	1,015.00
351	07/30/2012	1,377.82	328	10/11/2012	1,433.82	56.00	1,036.00

Summary for **SHORT** trades

78	Short trades opened in this time period				
33	Winners	Biggest Win	175.00	Sum of Gains	1,518.47
39	Losers	Biggest Loss	-35.00	Sum of Losses	-700.00
				NET change for Period	818.47
				Smallest Short Account Balance	-105.00

51 **Open Trade; Marked to current bar Open $**

Enter Short			Exit Short			Trade	Running
Rel Row	Date	$	Rel Row	Date	$	Result	Balance
8	07/30/2014	1,951.82	**OPEN**	08/08/2014	1,910.35	41.47	818.47
57	04/11/2014	1,846.82	51	04/17/2014	1,853.82	-7.00	777.00
67	03/31/2014	1,846.82	65	04/01/2014	1,867.82	-21.00	784.00
105	01/24/2014	1,818.82	91	02/10/2014	1,783.82	35.00	805.00
121	12/18/2013	1,776.82	119	12/19/2013	1,797.82	-21.00	770.00
157	10/01/2013	1,678.82	149	10/11/2013	1,685.82	-7.00	791.00
182	08/16/2013	1,671.82	171	09/10/2013	1,657.82	14.00	798.00
210	06/20/2013	1,615.82	201	07/05/2013	1,615.82	0.00	784.00
222	06/05/2013	1,629.82	214	06/18/2013	1,636.82	-7.00	784.00
245	04/23/2013	1,552.82	242	04/29/2013	1,580.82	-28.00	791.00
290	12/31/2012	1,405.82	289	01/02/2013	1,419.82	-14.00	819.00
322	10/23/2012	1,433.82	303	11/23/2012	1,391.82	42.00	833.00
327	10/16/2012	1,426.82	325	10/17/2012	1,447.82	-21.00	791.00
373	06/29/2012	1,335.82	372	06/29/2012	1,342.82	-7.00	812.00
403	05/11/2012	1,356.82	386	06/08/2012	1,314.82	42.00	819.00
420	04/10/2012	1,377.82	411	04/26/2012	1,377.82	0.00	777.00
465	12/19/2011	1,216.82	460	12/23/2011	1,244.82	-28.00	777.00
491	11/17/2011	1,230.82	478	11/30/2011	1,202.82	28.00	805.00

FIGURE 13: EXCEL, BACKTEST RESULTS. This shows the log of backtest transactions.

The spreadsheet file can be downloaded here: CreatingATradingStrategy.xlsm. To successfully download it, follow these steps:

- Right-click on the Excel file link ("CreatingATradingStrategy.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

TRADERS' TIPS

November 2014

For this month's Traders' Tips, the focus is mainly Sylvain Vervoort's article in this issue issue, "Price Projections" which is part 5 of his *Exploring Charting Techniques series*. Here we present the November 2014 Traders' Tips code with possible implementations in various software.

Code for NinjaTrader was already provided with Vervoort's article by the author. S&C subscribers will find that code at the Subscriber Area of our website here. (Click on "S&C Article Code" from the homepage.) 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: NOVEMBER 2014

In "Price Projections" in this issue, which is part 5 of an ongoing series titled *Exploring Charting Techniques*, author Sylvain Vervoort introduces several methods he uses for defining possible support & resistance levels. In the article, Vervoort provides code for his version of a daily pivot indicator that calculates support & resistance based on the prior day's high, low, and closing prices. We are providing TradeStation EasyLanguage code for a daily pivot indicator based on the calculations given in Vervoort's article.

To download the EasyLanguage code, please visit our TradeStation and EasyLanguage support forum. The code can be found at http://www.tradestation.com/TASC-2014. The ELD filename is "_TASC_SVEPivots.ELD."

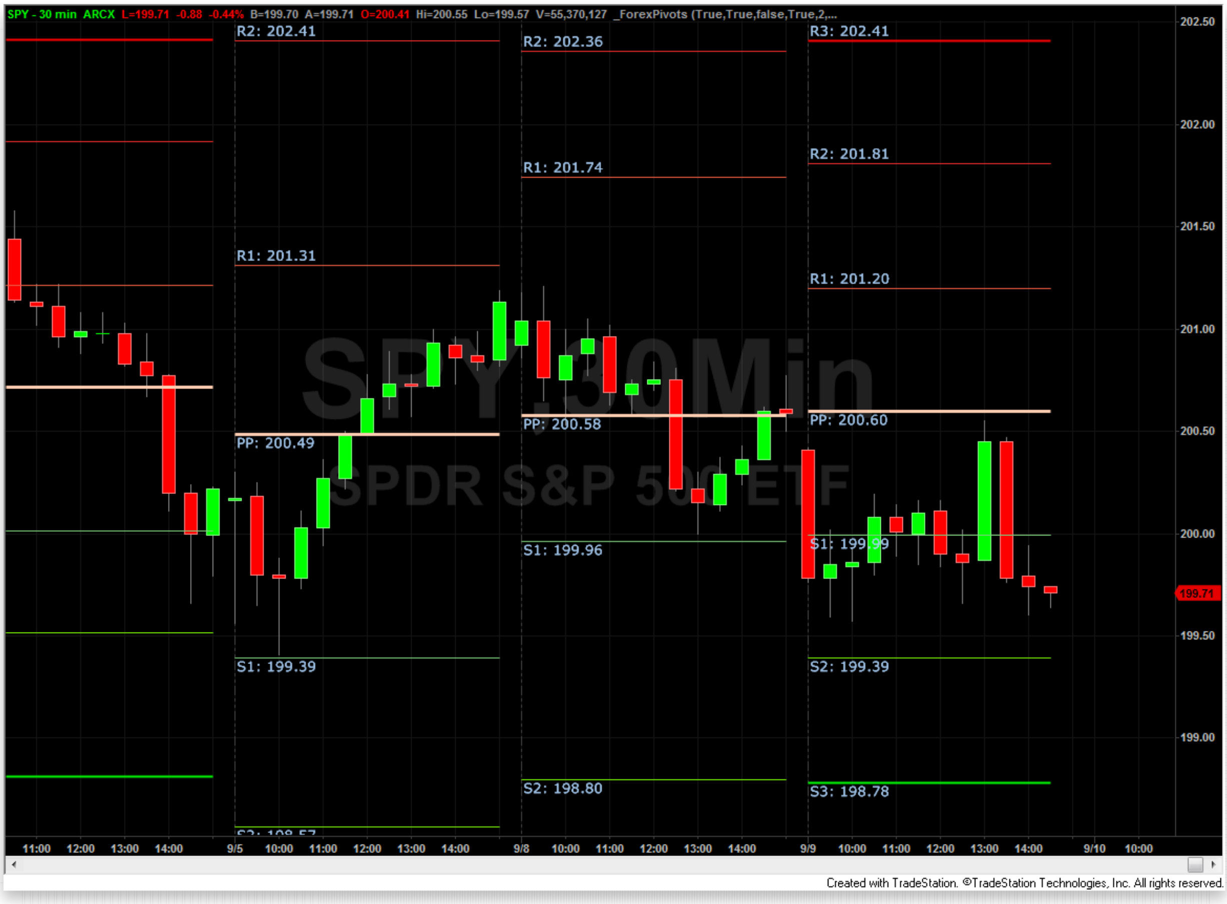
The code is also shown below.

_TASC_SVEPivots (Indicator)

```
{Reference: Technical Analysis of Stocks &amp; Commodities,
Nov. 2014.
Article: Price Projections
Indicator Name: _TASC_SVEPivots}


inputs:
        UseSessionBreak( True ),
        DrawTrendlines( True ),
        DrawPlots( False ),
        DrawTextBoxes( True ),
        PivotThickness( 2 ),
        PivotColor( RGB( 254, 204, 176 ) ),
        R1Color( RGB( 238, 92, 66 ) ),
        R2Color( RGB( 238, 44, 44 ) ),
        R3Color( RGB( 255, 0, 0 ) ),
        S1Color( RGB( 124, 205, 124 ) ),
        S2Color( RGB( 118, 238, 0 ) ),
        S3Color( RGB( 0, 255, 0 ) ),
        ShowR3( True ),
        ShowR2( True ),
        ShowR1( True ),
        ShowPP( True ),
        ShowS1( True ),
        ShowS2( True ),
        ShowS3( True ) ;

variables:
        MyPIP( 0 ),
        NumDecimals( 0 ),
        CS( 0 ),
    S1( 0 ),
        S2( 0 ),
        S3( 0 ),
        R1( 0 ),
        R2( 0 ),
        R3( 0 ),
        PP( 0 ),
        S1_ID( 0 ),
        S2_ID( 0 ),
        S3_ID( 0 ),
        R1_ID( 0 ),
        R2_ID( 0 ),
        R3_ID( 0 ),
        PP_ID( 0 ),
        S1_TextID( 0 ),
        S2_TextID( 0 ),
        S3_TextID( 0 ),
        R1_TextID( 0 ),
        R2_TextID( 0 ),
        R3_TextID( 0 ),
        PP_TextID( 0 ),
        TodaysHigh( 0 ),
        YestHigh( 0 ),
        TodaysLow( 0 ),
        YestLow( 0 ),
        TodaysClose( 0 ),
        YestClose( 0 ),
        CalcTrigger( false ),
        Counter( 0 ) ;

if CurrentBar = 1 then
        begin
        MyPIP = MinMove / PriceScale * 10 ;
        if Category = 12 then
                NumDecimals = Log( PriceScale ) / Log( 10 ) - 1
        else
                NumDecimals = Log( PriceScale ) / Log( 10 ) ;
        end ;

CS = CurrentSession( 0 ) ;

if UseSessionBreak then
        CalcTrigger = CS <> CS[1]
else
        CalcTrigger = Date <> Date[1] ;


if CalcTrigger then
        begin
        Counter = Counter + 1 ;
        YestHigh = TodaysHigh ;
        YestLow = TodaysLow ;
        YestClose = Close[1] ;
        TodaysHigh = High ;
        TodaysLow = Low ;
        PP = ( YestHigh + YestLow + YestClose ) / 3 ;
        R1 = PP * 2 - YestLow ;
        R2 = PP + YestHigh - YestLow ;
        R3 = ( 2*PP ) + ( YestHigh -( 2 * YestLow ) ) ;
        S1 = PP * 2 - YestHigh ;
        S2 = PP - YestHigh + YestLow ;
        S3 = ( 2 * PP ) - ( ( 2 * YestHigh ) - YestLow ) ;
        if DrawTrendlines then
                begin
                if ShowS1 then
                begin
                        S1_ID = TL_New( Date, Time, S1, Date, Time, S1 ) ;
                        TL_SetColor( S1_ID, S1Color ) ;
                        end ;
                if ShowS2 then
                begin
                        S2_ID = TL_New( Date, Time, S2, Date, Time, S2 ) ;
                        TL_SetColor( S2_ID, S2Color ) ;
                        end ;
                if ShowS3 then
                        begin
                        S3_ID = TL_New( Date, Time, S3, Date, Time, S3 ) ;
                        TL_SetColor( S3_ID, S3Color ) ;
                        TL_SetSize( S3_ID, 1 ) ;
                        end ;
                if ShowR1 then
                begin
                        R1_ID = TL_New( Date, Time, R1, Date, Time, R1 ) ;
                        TL_SetColor( R1_ID, R1Color ) ;
                        end ;
                if ShowR2 then
                begin
                        R2_ID = TL_New( Date, Time, R2, Date, Time, R2 ) ;
                        TL_SetColor( R2_ID, R2Color ) ;
                        end ;
                if ShowR3 then
                        begin
                        R3_ID = TL_New( Date, Time, R3, Date, Time, R3 ) ;
                        TL_SetColor( R3_ID, R3Color ) ;
                        TL_SetSize( R3_ID, 1 ) ;
                        end ;
                if ShowPP then
                begin
                        PP_ID = TL_New( Date, Time, PP, Date, Time, PP ) ;
                        TL_SetColor( PP_ID, PivotColor ) ;
                        TL_SetSize( PP_ID, PivotThickness ) ;
                        end ;
                end ;
        if DrawTextBoxes then
                begin
                if ShowS1 then S1_TextID = Text_New( Date, Time,
                        S1, "S1: " + NumToStr( S1, NumDecimals ) ) ;
                if ShowS2 then S2_TextID = Text_New( Date, Time,
                        S2, "S2: " + NumToStr( S2, NumDecimals ) ) ;
                if ShowS3 then S3_TextID = Text_New( Date, Time,
                        S3, "S3: " + NumToStr( S3, NumDecimals ) ) ;
                if ShowR1 then R1_TextID = Text_New( Date, Time,
                        R1, "R1: " + NumToStr( R1, NumDecimals ) ) ;
                if ShowR2 then R2_TextID = Text_New( Date, Time,
                        R2, "R2: " + NumToStr( R2, NumDecimals ) ) ;
                if ShowR3 then R3_TextID = Text_New( Date, Time,
                        R3, "R3: " + NumToStr( R3, NumDecimals ) ) ;
                if ShowPP then PP_TextID = Text_New( Date, Time,
                        PP, "PP: " + NumToStr( PP, NumDecimals ) ) ;
                if ShowS1 then Text_SetStyle( S1_TextID, 0, 0 ) ;
                if ShowS2 then Text_SetStyle( S2_TextID, 0, 0 ) ;
                if ShowS3 then Text_SetStyle( S3_TextID, 0, 0 ) ;
                if ShowPP then Text_SetStyle( PP_TextID, 0, 0 ) ;
                if ShowR1 then Text_SetStyle( R1_TextID, 0, 1 ) ;
                if ShowR2 then Text_SetStyle( R2_TextID, 0, 1 ) ;
                if ShowR3 then Text_SetStyle( R3_TextID, 0, 1 ) ;
                end ;
        end
else
        begin
        if High > TodaysHigh then
                TodaysHigh = High ;
        if Low < TodaysLow then
                TodaysLow = Low ;
        if DrawTrendlines then
                begin
                if S1_ID > 0 and ShowS1 and OneAlert( True ) then
                        TL_SetEnd( S1_ID, Date, Time, S1 ) ;
                if S2_ID > 0 and ShowS2 and OneAlert( True ) then
                        TL_SetEnd( S2_ID, Date, Time, S2 ) ;
                if S3_ID > 0 and ShowS3 and OneAlert( True ) then
                        TL_SetEnd( S3_ID, Date, Time, S3 ) ;
                if R1_ID > 0 and ShowR1 and OneAlert( True ) then
                        TL_SetEnd( R1_ID, Date, Time, R1 ) ;
                if R2_ID > 0 and ShowR2 and OneAlert( True ) then
                        TL_SetEnd( R2_ID, Date, Time, R2 ) ;
                if R3_ID > 0 and ShowR3 and OneAlert( True ) then
                        TL_SetEnd( R3_ID, Date, Time, R3 ) ;
                if PP_ID > 0 and ShowPP and OneAlert( True ) then
                        TL_SetEnd( PP_ID, Date, Time, PP ) ;
                end ;
        end ;

if DrawPlots then
        begin
        if Counter >= 2 and BarType < 3 then
                begin
                if ShowR3 then Plot1( R3, "R3"  ) ;
                if ShowR2 then Plot2( R2, "R2"  ) ;
                if ShowR1 then Plot3( R1, "R1"  ) ;
                if ShowPP then Plot4( PP, "PP"  ) ;
                if ShowS1 then Plot5( S1, "S1"  ) ;
                if ShowS2 then Plot6( S2, "S2"  ) ;
                if ShowS3 then Plot7( S3, "S3" ) ;
                end ;
        end ;
```

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

A sample chart implementing the indicator is shown in Figure 1.



Created with TradeStation. ©TradeStation Technologies, Inc. All rights reserved.

FIGURE 1: TRADESTATION. Here is a sample 30-minute chart of the SPY with the daily pivot indicator applied.

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: NOVEMBER 2014

For this month's Traders' Tip, we've provided the formula SVEPivotsUtcRt.efs based on the formula described in Sylvain Vervoort's article in this issue, "Price Projections" in his ongoing *Exploring Charting Techniques* series.

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 the study is shown in Figure 2.

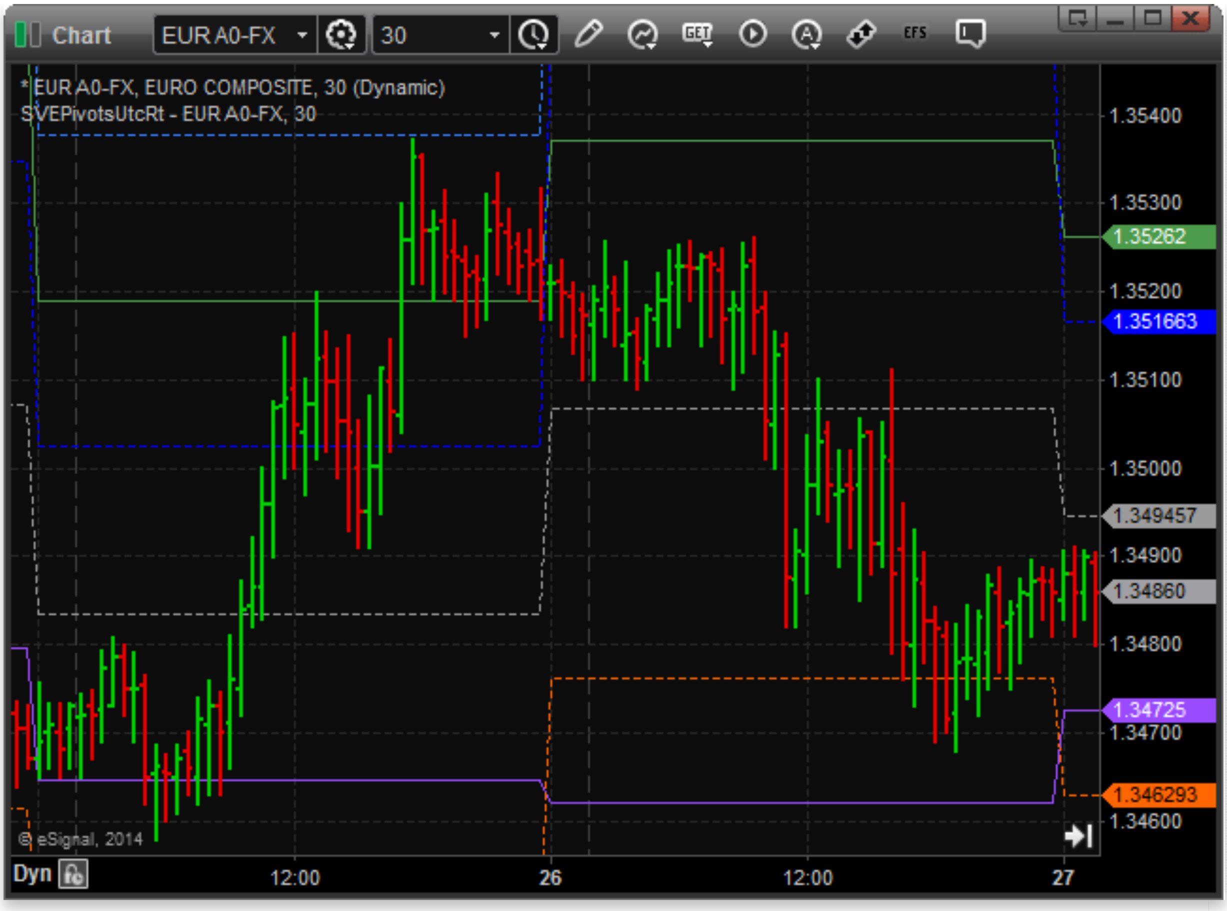


FIGURE 2: eSIGNAL. Here is an example of the study implemented on a 30-minute chart of the Euro Composite.

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:
    Price Projections by Sylvain Vervoort

Version:            1.00  09/04/2014

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

**********************************/
```

```
function preMain(){



    setPriceStudy(true);

    setStudyTitle("SVEPivotsUtcRt");



    setCursorLabelName("PH", 0);

    setCursorLabelName("PL", 1);

    setCursorLabelName("PP", 2);

    setCursorLabelName("R1", 3);

    setCursorLabelName("R2", 4);

    setCursorLabelName("R3", 5);

    setCursorLabelName("S1", 6);

    setCursorLabelName("S2", 7);

    setCursorLabelName("S3", 8);



    setDefaultBarFgColor(Color.RGB(0x4B,0x9B,0x4B), 0);
    setDefaultBarFgColor(Color.RGB(0x9B,0x4B,0xFF), 1);
    setDefaultBarFgColor(Color.RGB(0x9B,0x9B,0x9B), 2);
    setDefaultBarFgColor(Color.RGB(0x00,0x00,0xFF), 3);
    setDefaultBarFgColor(Color.RGB(0x32,0x7D,0xFF), 4);

    setDefaultBarFgColor(Color.RGB(0x00,0xFF,0xFF), 5);
    setDefaultBarFgColor(Color.RGB(0xFF,0x65,0x00), 6);
    setDefaultBarFgColor(Color.RGB(0xFF,0x94,0x00), 7);
    setDefaultBarFgColor(Color.RGB(0xFF,0xC8,0x00), 8);



    setDefaultBarStyle(PS_SOLID, 0);
    setDefaultBarStyle(PS_SOLID, 1);
    setDefaultBarStyle(PS_DASH, 2);
    setDefaultBarStyle(PS_DASH, 3);
    setDefaultBarStyle(PS_DASH, 4);

    setDefaultBarStyle(PS_DASH, 5);
    setDefaultBarStyle(PS_DASH, 6);
    setDefaultBarStyle(PS_DASH, 7);
    setDefaultBarStyle(PS_DASH, 8);
}
```

```
var bInit = false;

var bVersion = null;


var xHigh  = null;

var xLow   = null;

var xClose = null;


function main(){


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


    if(isMonthly() || isWeekly())

        return;


    if(bInit == false){

        xHigh  = high(inv("D"));

        xLow   = low(inv("D"));

        xClose = close(inv("D"));

        bInit = true;

    }


    var vHigh  = xHigh.getValue(-1);

    var vLow   = xLow.getValue(-1);

    var vClose = xClose.getValue(-1);


    if(vHigh == null || vLow == null || vClose == null)

        return;


    var vPP = (vHigh + vLow + vClose) / 3;

    var vR1 = 2*vPP - vLow;
    var vS1 = 2*vPP - vHigh;

    var vR2 = vPP + (vHigh - vLow);

    var vS2 = vPP - (vHigh - vLow);

    var vR3 = 2*vPP + (vHigh - 2*vLow);

    var vS3 = 2*vPP - (2*vHigh - vLow);



    return [vHigh, vLow, vPP, vR1, vR2, vR3, vS1, vS2, vS3];

}



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: NOVEMBER 2014

In his article in this issue, “Price Projections,” which is part 5 of his ongoing series on exploring charting techniques, author Sylvain Vervoort introduces his SVEPivotsUtcRt indicator as a way of drawing daily pivots on the chart. In today’s trading platforms, the ability draw pivots is commonplace, and Wealth-Lab is no exception. Thus, we don’t need to provide any custom code.

After you download all the available strategies using the *download* button in the *open strategy* dialog, look for the strategy named “floor trader pivots.” Likewise, the other techniques mentioned by Vervoort — namely, the step candle pattern and 1-2-3 wave count — are also ready to be explored as downloadable strategies in Wealth-Lab.

A sample chart showing some *floor trader pivots* is in Figure 3.

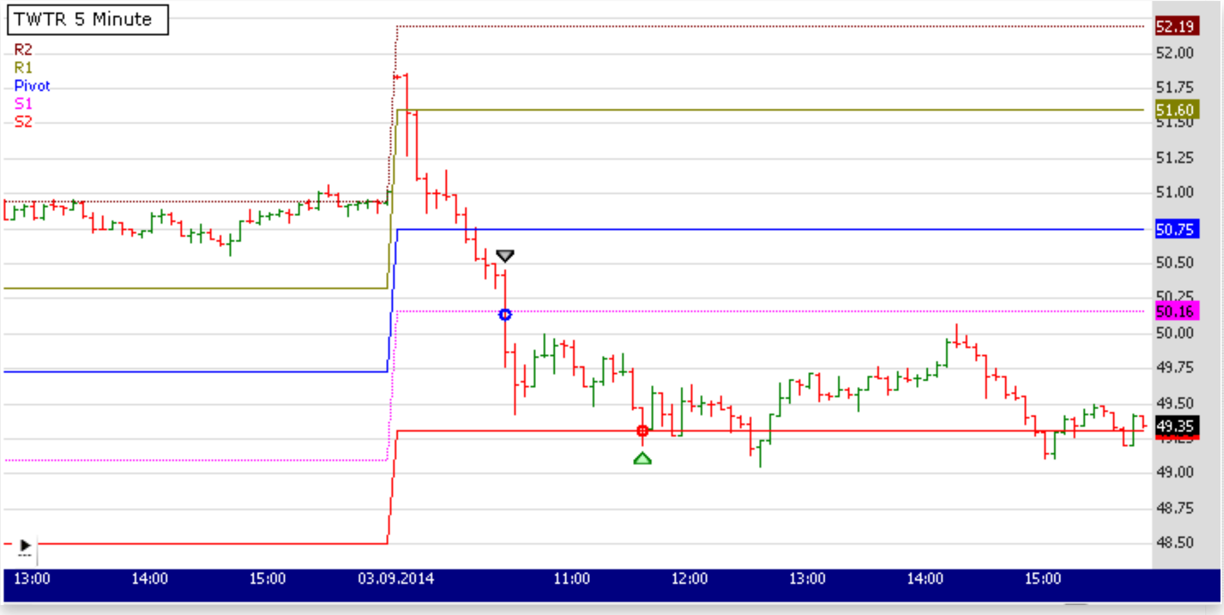


FIGURE 3: WEALTH-LAB, PIVOTS. Here is a sample Wealth-Lab 6 chart illustrating the application of the floor trader pivots trading system on a five-minute chart of Twitter (TWTR).

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

BACK TO LIST

NEUROSHELL TRADER: NOVEMBER 2014

In “Price Projections” in this issue, author Sylvain Vervoort discusses more elements of charting in his ongoing series on exploring charting techniques, including measured moves, Fibonacci projections & retracements, and daily pivots. Here, we’ll take a look at how some of the charting techniques he describes can be invoked in NeuroShell Trader.

Price levels and projections can be easily implemented in NeuroShell Trader using the following methods:

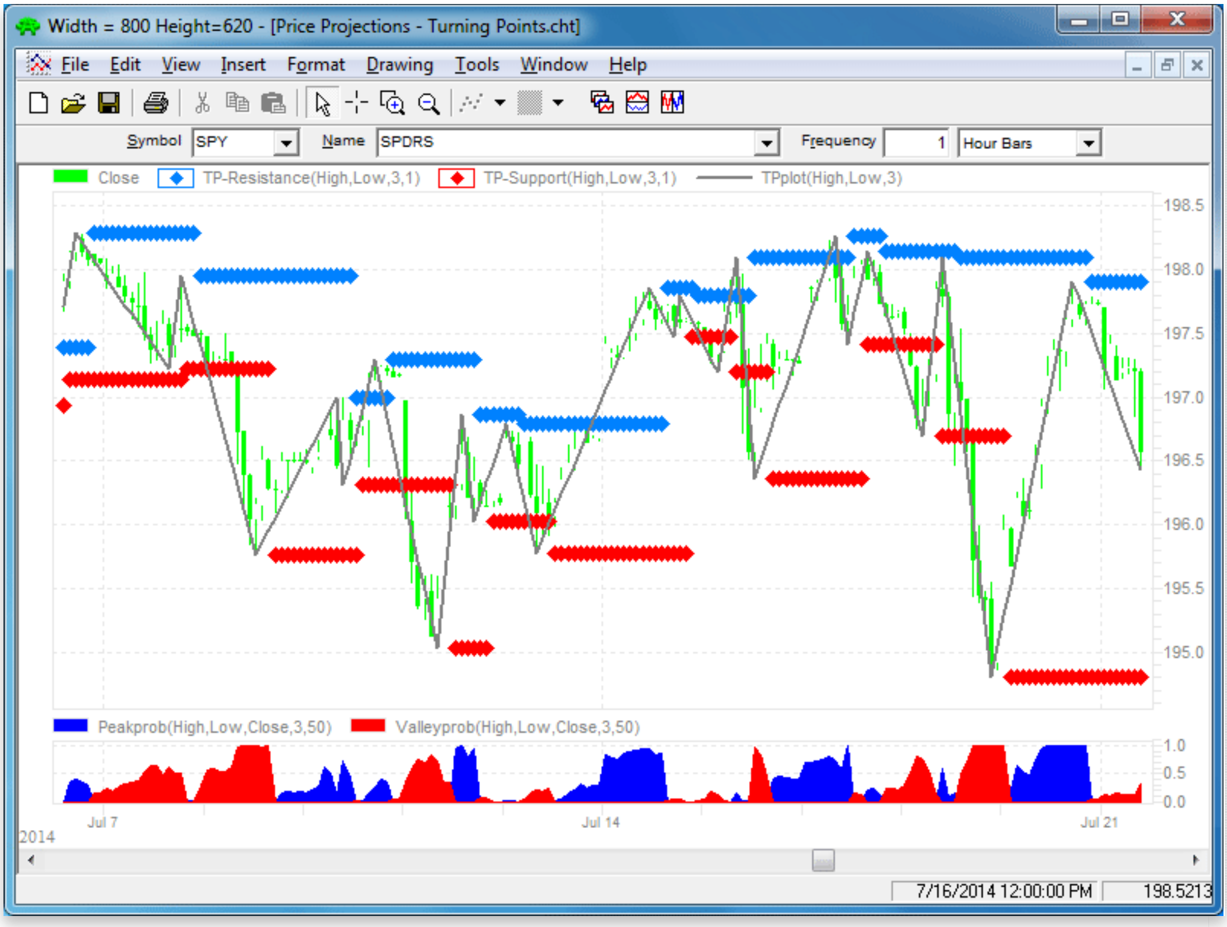
1. ***Turning points.*** The *Turning Points* add-on for NeuroShell Trader helps find local peaks & valleys in a price series. It allows implementation of price swings and projections into automated trading systems. Among other things, the turning points indicator computes support & resistance lines from prior price swings; Fibonacci retracement lines from each price swing; and the probability that the current price level is at a new turning point based on statistical measures.
2. ***Past price swings.*** Sylvain Vervoort’s SVEHLZZperc indicator, which he introduced in his June 2013 STOCKS & COMMODITIES article “The 1-2-3 Wave Count,” uses the zigzag indicator to identify past price swings on the chart. This indicator is available for download from our support site www.ward.net.
3. ***Fibonacci levels.*** You can draw Fibonacci retracements, Fibonacci projections, and even Fibonacci timelines on the chart using NeuroShell Trader’s built-in drawing tools.
4. ***Pivot points.*** Create *daily pivot* indicators using 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:

```
Pivot Point PP:  Avg3(DayHigh(High,1), DayLow(Low,1), DayClose(Close,1))
Resistance R1:  Subtract(Add2(PP, PP), DayLow(Low,1))
Resistance R2:  Add2(PP, DayRange(High,Low1))
Resistance R3:  Add2(R1, DayRange(High,Low1))
Support S1:  Subtract(Add2(PP, PP), DayHigh(High,1))
Support S2:  Subtract(PP, DayRange(High,Low1))
Support S3:  Subtract(S1, DayRange(High,Low1))
```

5. ***Projected price swings.*** Use NeuroShell Trader’s built-in neural network predictions to find patterns in past data and then use those patterns to identify likely future price swings.

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.



Width = 800 Height=620 - [Price Projections - Turning Points.cht]

Symbol SPY Name SPDRS Frequency 1 Hour Bars

7/16/2014 12:00:00 PM 198.5213

FIGURE 4: NEUROSHELL TRADER. This sample NeuroShell Trader chart displays a few of the turning point add-on indicators.

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

BACK TO LIST

METASTOCK: NOVEMBER 2014

Sylvain Vervoort's article in this issue, "Price Projections," includes his version of daily pivots. The following MetaStock formula is designed to change "days" when the Coordinated Universal Time (UTC or GMT) shows midnight. The formula prompts the user to enter his time zones UTC adjustment. If the indicator is being plotted on a daily or higher interval, the UTC adjustment is not used.

```
rollmod:= Input("Coordinated Universal Time (or UTC) adjustment", -12, 13, -5);
rolltime:= If(rollmod >= 0, rollmod, 24-rollmod);
roll:= If(rolltime = 0, Hour() < Ref(Hour(),-1),
    Hour() < rolltime AND Hour() >= rolltime);
intraday:= LastValue(Max(Cum(Hour())<>0, Cum(Minute())<>0) >0);
new:=If(intraday, roll, ROC(DayOfWeek(),1,$)<>0);
yh:=ValueWhen(1,new, Ref(HighestSince(1,new,H),-1));
yl:=ValueWhen(1,new, Ref(LowestSince(1,new,L),-1));
yc:=ValueWhen(1,new, Ref(C,-1));

pp:=(yc+yh+yl)/3;
r1:=(pp*2)-yl;
s1:=(pp*2)-yh;
r2:= pp+r1-s1;
s2:= pp-r1+s1;
r3:= pp+r2-s2;
s3:= pp-r2+s2;

r3;
r2;
r1;
pp;
s1;
s2;
s3;
```

—William Golson
MetaStock Technical Support
www.metastock.com

BACK TO LIST

AMIBROKER: NOVEMBER 2014

In “Price Projections” in this issue, author Sylvain Vervoort continues his article series on charting techniques, including using daily pivot points to estimate future price levels.

A ready-to-use AmiBroker formula for daily pivots is presented here. A sample chart is shown in Figure 5.

```
LISTING 1.
ph = TimeFrameGetPrice("H", inDaily, -1 );
pl = TimeFrameGetPrice("L", inDaily, -1 );
pc = TimeFrameGetPrice("C", inDaily, -1 );

PP = ( PH + PL + PC )/3;

R1 = 2 * PP - PL;
R2 = PP + PH - PL;
R3 = R1 + PH - PL;

S1 = 2 * PP - PH;
S2 = 2 * PP - PH - PL;
S3 = S1 - PH - PL;
```

```
Plot( C, "Price", colorDefault, styleBar | styleThick );

Plot( PH, "PH", colorGreen, styleNoRescale );
Plot( PL, "PL", colorViolet, styleNoRescale );
Plot( PP, "PP", colorBlack, styleNoRescale );
Plot( R1, "R1", colorBlue, styleDashed | styleNoRescale );
Plot( R2, "R2", colorLightBlue, styleDashed | styleNoRescale );
Plot( R3, "R3", colorAqua, styleDashed | styleNoRescale );
Plot( S1, "S1", colorRed, styleDashed | styleNoRescale );
Plot( S2, "S2", colorOrange, styleDashed | styleNoRescale );
Plot( S3, "S3", colorDarkYellow, styleDashed | styleNoRescale );
```

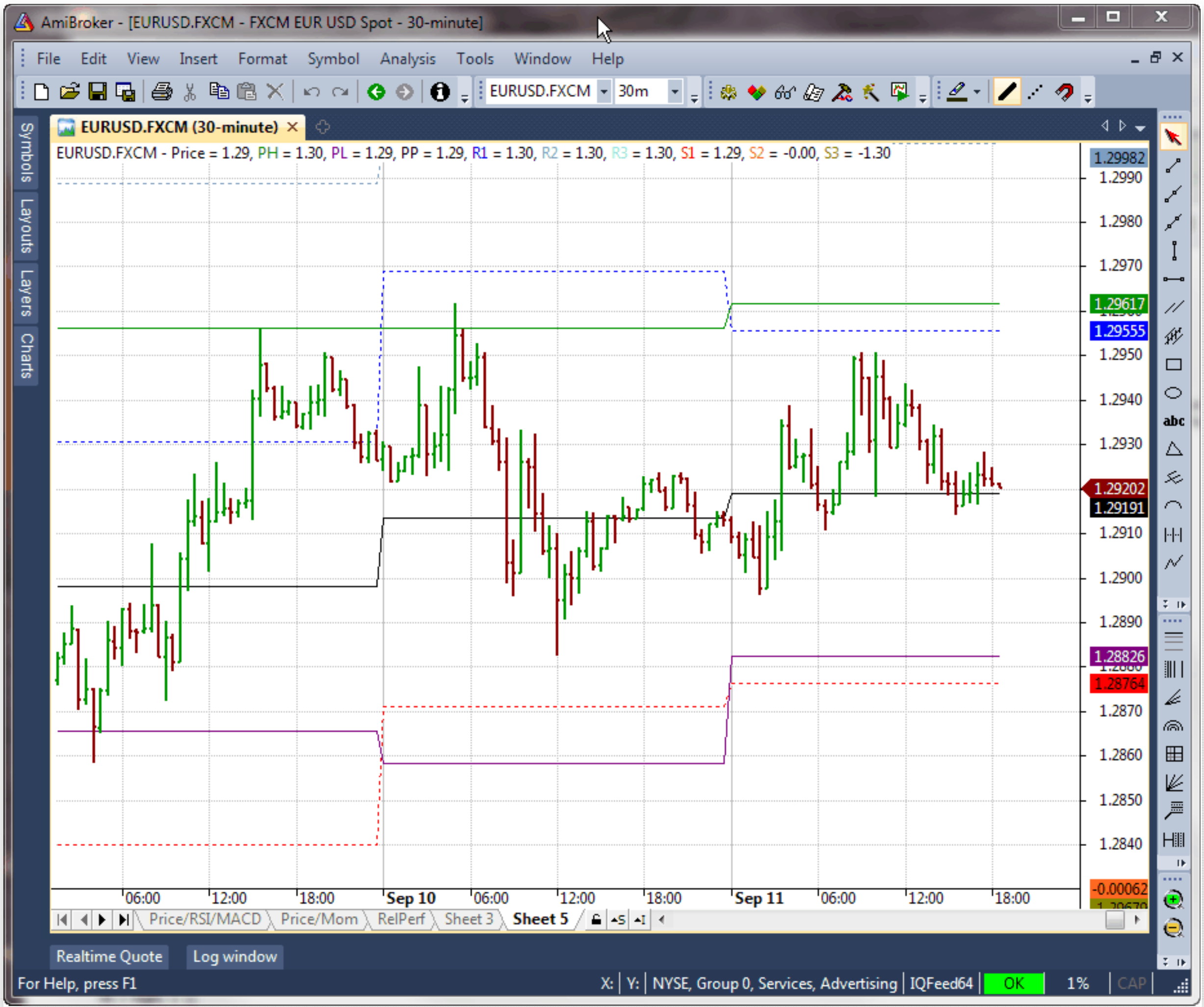


FIGURE 5: AMIBROKER. Here is a sample EURUSD 30-minute chart with daily pivots and support/resistance levels shown.

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

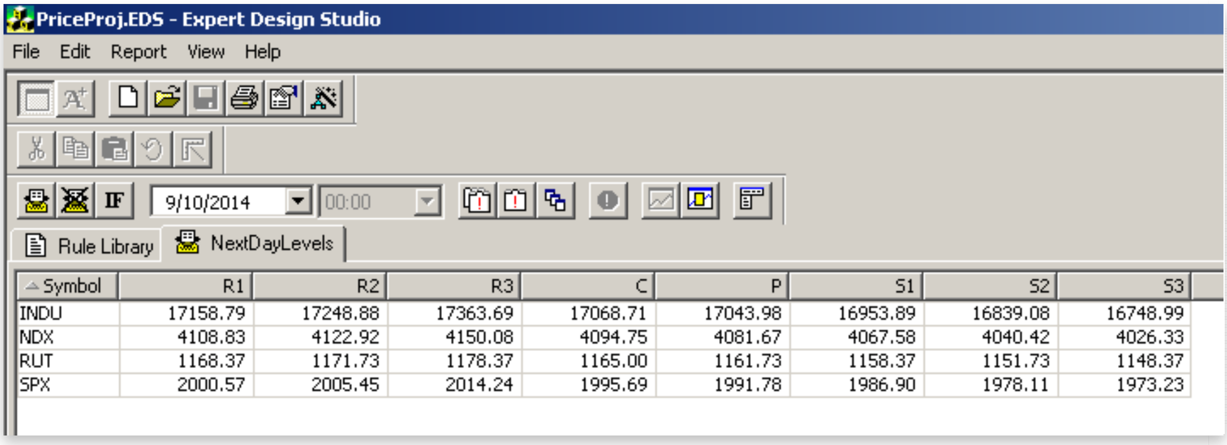
BACK TO LIST

AIQ: NOVEMBER 2014

The AIQ code for this month is based on Sylvain Vervoort's article in this issue, "Price Projections," which is part 5 of his ongoing series on exploring charting techniques. The AIQ code and EDS file can be downloaded from www.TradersEdgeSystems.com/traderstips.htm.

The code runs on daily bars only and computes the various support & resistance levels for the next day's intraday trading. The levels cannot be plotted on the *real-time alerts* chart.

In Figure 6, I show a report that was run on the major indexes for 9/10/2014.



Symbol	R1	R2	R3	C	P	S1	S2	S3
INDU	17158.79	17248.88	17363.69	17068.71	17043.98	16953.89	16839.08	16748.99
NDX	4108.83	4122.92	4150.08	4094.75	4081.67	4067.58	4040.42	4026.33
RUT	1168.37	1171.73	1178.37	1165.00	1161.73	1158.37	1151.73	1148.37
SPX	2000.57	2005.45	2014.24	1995.69	1991.78	1986.90	1978.11	1973.23

FIGURE 6: AIQ. Support & resistance levels are calculated based on end-of-day data as of 9/10/2014 for the major indexes. The levels are for use in intraday trading for the next day (9/11/2014).

The code is as follows:

```
!PRICE PROJECTIONS
!Author: Sylvain Vervoort, TASC Nov 2014
!Coded by: Richard Denning 9/10/2014
!www.TradersEdgeSystems.com

C is [close].
H is [high].
L is [low].

!To get next day levels we will be running the report at the end of day
!so prior high will be the current H, etc. The report will give the values for
!the next day but cannot be ploted on a real time chart since we are using
```

```
!daily end of day data to compute the levels.

!THIS CODE RUNS ON DAILY DATA:
P is (H+L+C)/3.
R1 is (2*P) - L.
R2 is P + (H - L).
R3 is (2*P) + (H - (2*L)).
S1 is (2*P) - H.
S2 is P - (H - L).
S3 is (2*P) - ((2*H) - L).

NextDayLevels if C > 0 and H > 0
        and L > 0 and H - L > 0.
```

—Richard Denning
info@TradersEdgeSystems.com
for AIQ Systems

BACK TO LIST

TRADERSSTUDIO: NOVEMBER 2014

The TradersStudio code based on Sylvain Vervoort's article in this issue, "Price Projections," 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 PIVOTS:** Returns the pivot point and all the support & resistance levels based on daily high, low, and close of the input data. This is the value for the next day if the input is the current end-of-day close
- **Indicator plot PIVOTS_IND:** Plots the six support & resistance levels from the PIVOTS function on the next day's intraday bars shifted back one day
- **System DISPLAY_PIVOTS:** This is not a trading system but just my way of displaying the levels on a chart.

The support & resistance levels can be displayed on any historical intraday chart. The code uses the same data file compressed to daily data by setting it up as the child datastream with the *type* set to "daily." Note that this cannot be used to trade intraday, as TradersStudio does not as of yet have a real-time module.

In Figure 7, I show a chart of the emini futures contract (ES) 60-minute bars with the support & resistance levels calculated from yesterday's daily bar data.

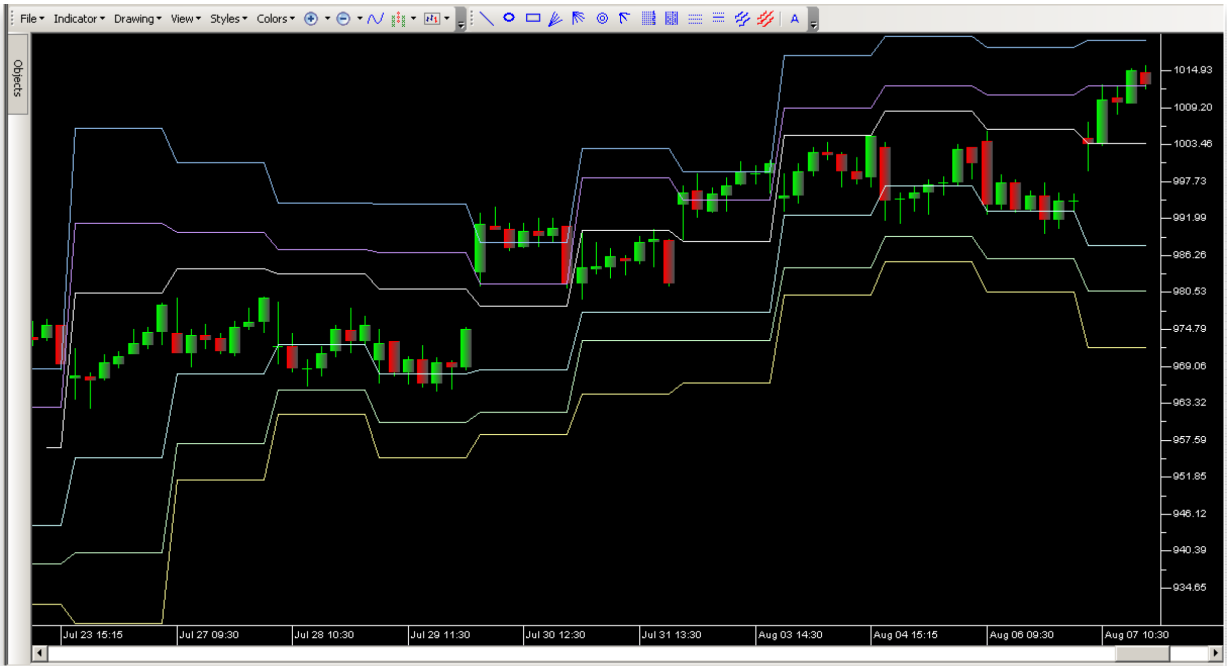


FIGURE 7: TRADERSSTUDIO, SUPPORT & RESISTANCE. Here is a sample chart of the emini futures contract (ES) 60-minute bars with the support & resistance levels calculated from yesterday's daily bar data.

The code is as follows:

```
'PRICE PROJECTIONS
'Author: Sylvain Vervoort, TASC Nov 2014
'Coded by: Richard Denning 9/10/2014
'www TradersEdgeSystems.com

Function PIVOTS(PriceH as bararray,PriceL as bararray,PriceC as bararray,ByRef
P,ByRef R1,ByRef R2, ByRef R3, ByRef S1, ByRef S2, ByRef S3)
'THIS CODE RUNS ON DAILY DATA AND COMPUTES THE NEXT DAYS SUPPORT AND RESISTANCE
LEVELS:
'Dim P As BarArray
P = (PriceH+PriceL+PriceC)/3
R1 = (2*P) - PriceL
R2 = P + (PriceH - PriceL)
R3 = (2*P) + (PriceH - (2*PriceL))
S1 = (2*P) - PriceH
S2 = P - (PriceH - PriceL)
S3 = (2*P) - ((2*PriceH) - PriceL)
PIVOTS = P
End Function
'-----------------------------------------------------------------------------------------
------------------------------------------------------
'INDICATOR PLOT CODE:
Sub PIVOTS_IND()
Dim priceH As BarArray
Dim priceL As BarArray
Dim priceC As BarArray
Dim P As BarArray
Dim R1 As BarArray
Dim R2 As BarArray
Dim R3 As BarArray
Dim S1 As BarArray
Dim S2 As BarArray
Dim S3 As BarArray

priceH = H Of independent1
priceL = L Of independent1
priceC = C Of independent1

P = PIVOTS(priceH,priceL,priceC,P,R1,R2,R3,S1,S2,S3)

plot1(R1[1])
plot2(R2[1])
plot3(R3[1])
plot4(S1[1])
plot5(S2[1])
plot6(S3[1])

End Sub
'-------------------------------------------------------------------------------------
-------------------------------------------------------
sub DISPLAY_PIVOTS()
Dim DailyData As BarArray
DailyData = C Of independent1
If BarNumber  > BarSize Then Buy("LE",1,0,Market,Day)
If BarNumber  = LastBar - 1 Then ExitLong("LX","",1,0,Market,Day)

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

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

UPDATA: NOVEMBER 2014

Our Traders' Tip for this month is based on the article "Price Projections" in this issue by Sylvain Vervoort, which is the fifth part of his ongoing series, *Exploring Charting Techniques*.

In the article, Vervoort defines a set of pivot points, which are supposed places of support or resistance, based on ratios of the previous day's high, low and close. These levels are then plotted over the current day's price.

Hardcoded versions of Fibonacci levels already exist in Updata, so we don’t need to provide custom code for those.

The Updata code for Vervoort’s technique of finding pivot points has been introduced into the Updata Library. You can download the code 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.

```
'Pivot Points

NAME "Pivots" ""
DISPLAYSTYLE 7LINES
INDICATORTYPE TOOL
PLOTSTYLE THICK2 RGB(0,0,255)
PLOTSTYLE2 LINE RGB(150,150,150)
PLOTSTYLE3 LINE RGB(150,150,150)
PLOTSTYLE4 LINE RGB(150,150,150)
PLOTSTYLE5 LINE RGB(150,150,150)
PLOTSTYLE6 LINE RGB(150,150,150)
PLOTSTYLE7 LINE RGB(150,150,150)
@RUNNINGHIGH=0
@RUNNINGLOW=0
@THISDAYOPEN=0
@LASTDAYCLOSE=0
@LASTDAYHIGH=0
@LASTDAYLOW=0
@LASTDAYOPEN=0
@PivotPoint=0
@R1=0
@R2=0
@R3=0
@S1=0
@S2=0
@S3=0
FOR #CURDATE=0 TO #LASTDATE
     If #CURDAY!=HIST(#CURDAY,1)
         @LASTDAYCLOSE=HIST(CLOSE,1)
         @LASTDAYHIGH=HIST(@RUNNINGHIGH,1)
         @LASTDAYLOW=HIST(@RUNNINGLOW,1)
         @LASTDAYOPEN=HIST(@THISDAYOPEN,1)
         'At the start of a new day, initialise OHL values
         @RUNNINGHIGH=HIGH
         @RUNNINGLOW=LOW
         @THISDAYOPEN=OPEN
         @PLOT=-10000
         @PLOT2=-10000
         @PLOT3=-10000
         @PLOT4=-10000
         @PLOT5=-10000
         @PLOT6=-10000
         @PLOT7=-10000
     Else
         If HIGH>@RUNNINGHIGH
            @RUNNINGHIGH=HIGH
         ElseIf LOW<@RUNNINGLOW
            @RUNNINGLOW=LOW
         EndIf
         @PivotPoint=(@LASTDAYHIGH+@LASTDAYLOW+@LASTDAYCLOSE)/3
         @R1=2*@PivotPoint-@LASTDAYLOW
         @S1=2*@PivotPoint-@LASTDAYHIGH
         @R2=@PivotPoint+(@LASTDAYHIGH-@LASTDAYLOW)
         @S2=@PivotPoint-(@LASTDAYHIGH-@LASTDAYLOW)
         @R3=(2*@PivotPoint)+(@LASTDAYHIGH-(2*@LASTDAYLOW))
         @S3=2*@PivotPoint-((2*@LASTDAYHIGH)-@LASTDAYLOW)
         @PLOT=@PivotPoint
         @PLOT2=@S1
         @PLOT3=@R1
         @PLOT4=@S2
         @PLOT5=@R2
         @PLOT6=@S3
         @PLOT7=@R3
     EndIf
NEXT
```

A sample figure is shown in Figure 8.

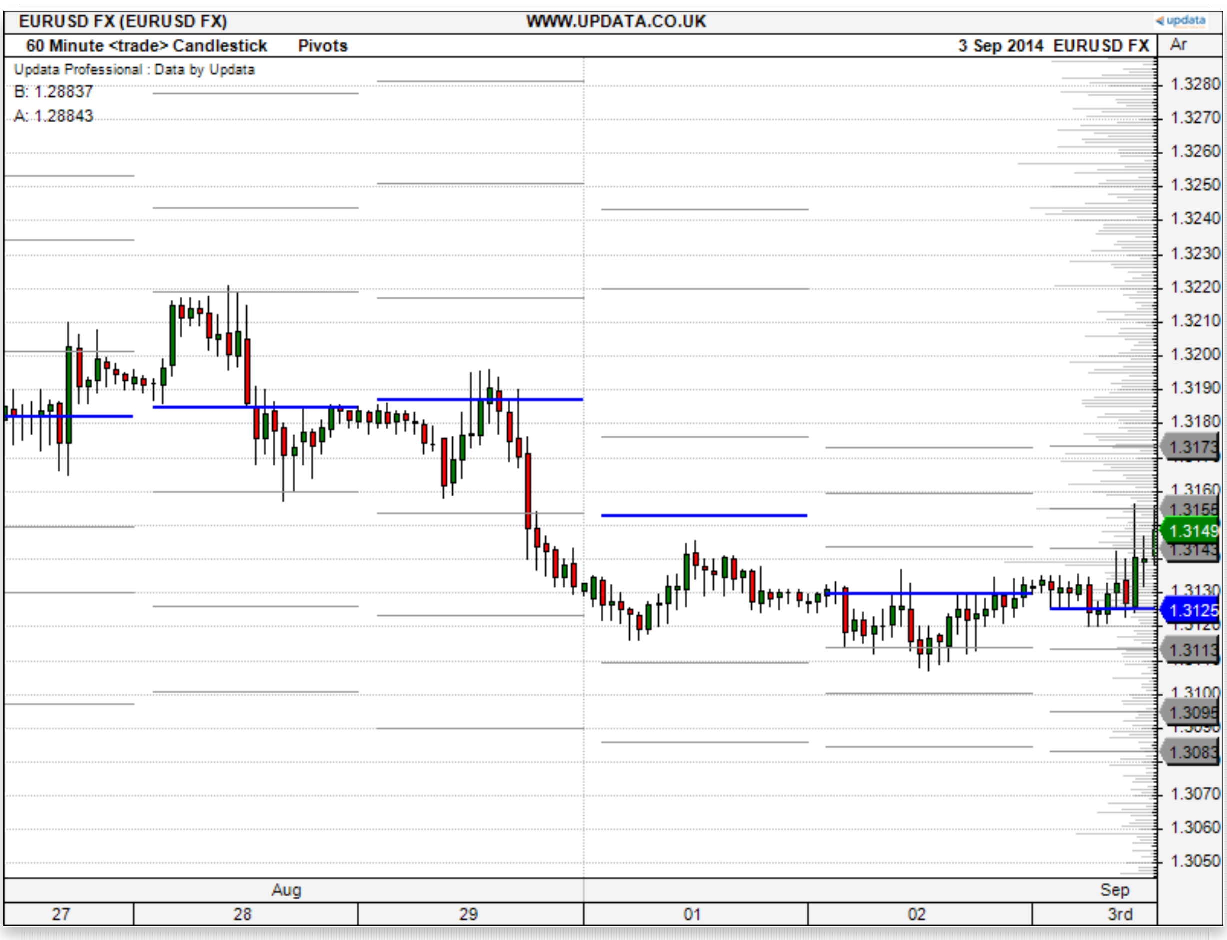


FIGURE 8: UPDATA, PIVOT POINTS. Here is an example of Sylvain Vervoort's described technique for finding pivot points applied to the currency pair EUR/USD in 60-minute resolution.

—Updata support team

support@updata.co.uk
www.updata.co.uk

BACK TO LIST

NINJATRADER: NOVEMBER 2014

Since NinjaScript code for NinjaTrader was already given in Sylvain Vervoort's article in this issue ("Price Projections") for his techniques in this month's installment of his "Exploring Charting Techniques" series, we'll focus here on another topic: the *choppiness index*.

The choppiness index was designed to help determine whether the market is choppy (that is, trading sideways) or not (trading with a trend in either direction). This study is not meant to predict the future market direction, but rather, it's just a metric to be used for defining the market's trendiness. Higher values equate to more choppiness, while lower values signify more directional trending character in price.

However, it can be hard to determine the exact trend in which the market is making with the lower values. With this in mind, we have included additional checks for when our values are falling and our closing prices start to go below or above a moving average. If the market is trending up, we will see a green study plot; if down, a red one.

This indicator is available for download at www.ninjatrader.com/SC/November2014SC.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 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 "ChopIndicator" file.

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

A sample chart implementing the choppiness index is shown in Figure 9.

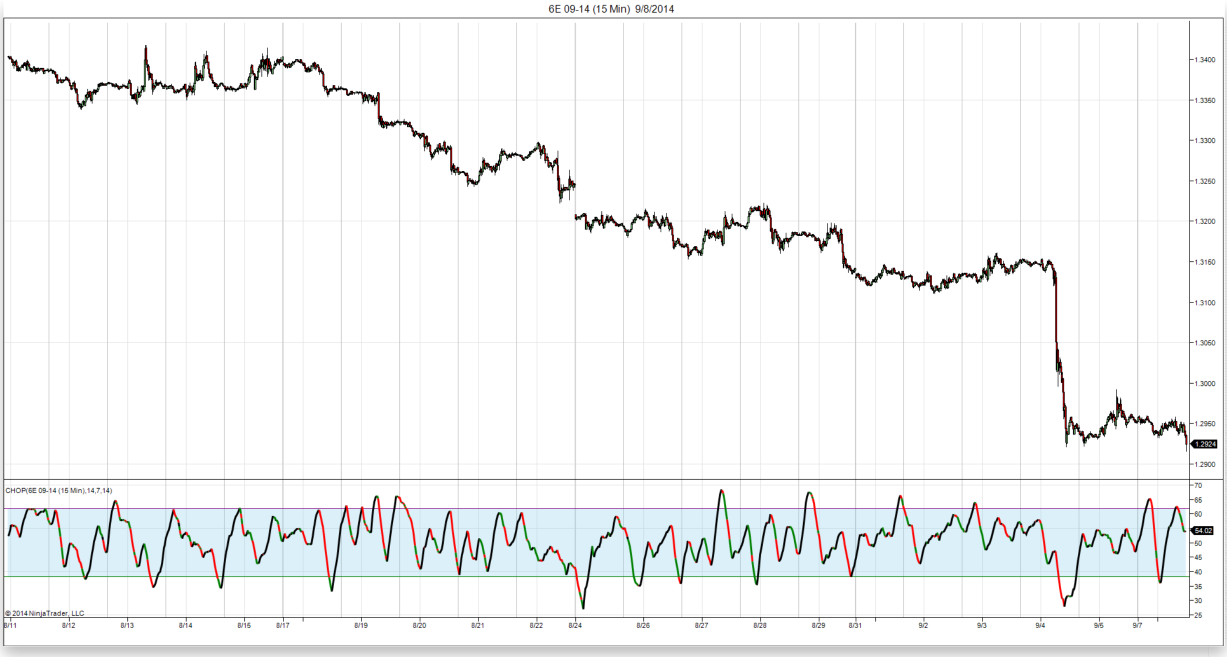


FIGURE 9: NINJATRADER, CHOPPINESS INDEX. This screenshot shows the indicator applied to a 15-minute EuroFX futures chart in NinjaTrader.

—Raymond Deux & Cal Hueber
NinjaTrader, LLC
www.ninjatrader.com

BACK TO LIST

December 2014

For this month's Traders' Tips, the focus is Markos Katsanos' article in this issue, "Detecting Flags In Intraday Charts." Here we present the December 2014 Traders' Tips code with possible implementations in various software.

Code for TradeStation was already provided in Katsanos' article by the author. S&C subscribers will find that code at the Subscriber Area of our website, www.traders.com. 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: DECEMBER 2014

In "Detecting Flags In Intraday Charts" in this issue, author Markos Katsanos describes a method for identifying chart patterns known as *flags* on intraday charts. The author provides some TradeStation strategy code for a trading system based on his rules. Here, we are providing some additional EasyLanguage code for TradeStation for an indicator based on the same rules. The indicator can be used with a chart as well as with the TradeStation Scanner to search your symbol list of stocks.

To download this EasyLanguage code, please visit our TradeStation and EasyLanguage support forum. The code can be found here: http://www.tradestation.com/TASC-2014. The ELD filename is "_TASC_DEC2014_INTRADAYFLAGS.ELD".

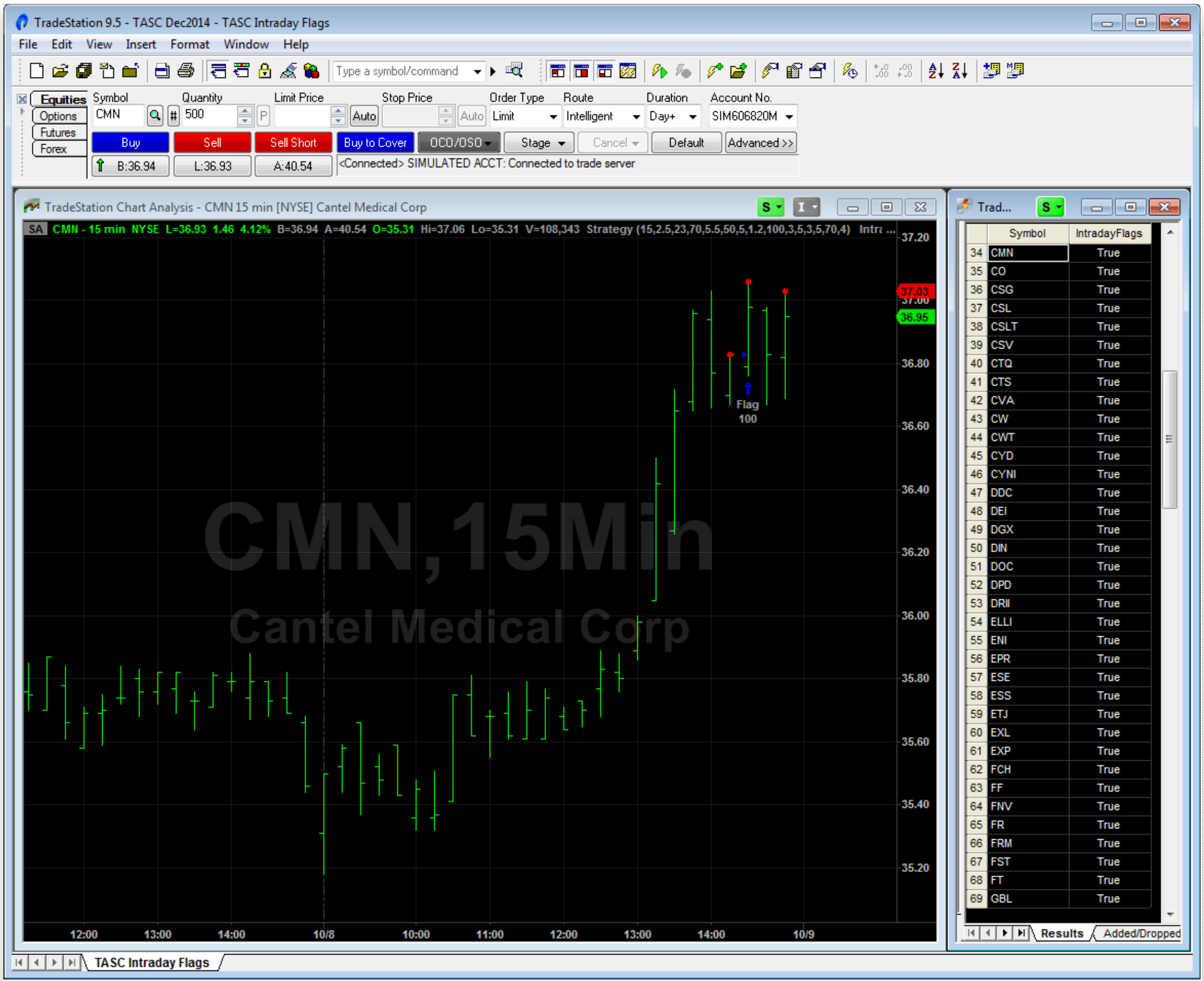
The code is also shown here:

```
IntradayFlags (Indicator)
{
        TASC December 2014
        Detecting Flags in Intraday Charts
        by Markos Katsanos
        Indicator
}

inputs:
        MAXFLDUR( 15 ), //Max Flag Duration
        FLAGMIN( 2.5 ), // Max ATR in lowest point in flag
        PX( 23 ), //Max Pole Duration.
        UPT1BARS( 70 ), // Bars for Uptrend leading to flag
        POLEMIN( 5.5 ), //Min ATR Height of the pole
        ATRmin( 5 ) ;// Min volatility change


variables:
        X1( 0 ),
        X2( 0 ),
        LRSX2( 0 ),
        LRSX1( 0 ),
        POLE( 0 ),
        TOP( 0 ),
        BOTTOM(  0),
        Y23( 0 ),
        FLAGBOT( 0 ),
        UPT1( 0 ),
        LF( 0 ),
        TARGETPER( 0 ) ;


// FLAG CALCULATION
//Flag duration ex pole top
X1 = HighestBar( Close, MAXFLDUR )[2] ;
//Flag duration including pole top
X2 = X1 + 1 ;
LF = Lowest( Close, X2 ) ;
TOP = Highest( Close, X2 )[2] ;
X2=Iff( LinearRegSlope( Close, X1 )[1] < 0
        and TOP - LF < flagmin * AvgTrueRange( 40 ), X1 + 1, 100 ) ;

if X2 > 2 and X2 <= MAXFLDUR then
        begin //Limits flag duration
        Y23=LowestBar( Close, PX + X2 ) ; // Pole bottom bar
        BOTTOM=Lowest( Close, PX+X2 ) ;// Pole bottom
        POLE = TOP - BOTTOM;
        IF TOP - BOTTOM > POLEMIN * AvgTrueRange( 40 )
                and Y23 > X2 then
                begin
                TOP = Highest( Close, X2 )[2] ;
                FLAGBOT = Lowest( Close, X2 ) ;
                //Uptrend leading to flag
                UPT1 = BOTTOM - Lowest( Low, UPT1BARS ) ;
                //Slope in flag
                LRSX1 = LinearRegSlope( Close, X1 ) * 100 ;
                //Slope in flag before breakout
                LRSX2 = LinearRegSlope( Close, X1-1 )[2] * 100 ;
                //Limits flag slope between 0 and -3 ATR
                Condition1 = TOP-LF < flagmin * AvgTrueRange( 40 )
                        and ( LRSX1 < 0 or LRSX2 < 0 ) ;
                //Limits min pole height
                Condition2 = POLE > POLEMIN * AvgTrueRange( 40 ) ;
                //Uptrend leading to flag
                Condition3 = UPT1 > 0 ;
                //Volatility
                Condition5 = ( AvgTrueRange( 40 ) /
                        AvgTrueRange( 40 )[Y23] - 1 ) * 100 > ATRmin ;
                If Condition1 and Condition2 and Condition3
                        and Condition5 then
                        begin
                        Plot1( High, "Flag" ) ;
                        Alert ;
                        end ;
                end ;
        end ;
```

```
IntradayFlags (Strategy)
{
        TASC December 2014
        Detecting Flags in Intraday Charts
        by Markos Katsanos
        Strategy
}

inputs:
        MAXFLDUR( 15 ), //Max Flag Duration
        FLAGMIN( 2.5 ), // Max ATR in lowest point in flag
        PX( 23 ), //Max Pole Duration.
        UPT1BARS( 70 ), // Bars for Uptrend leading to flag
        POLEMIN( 5.5 ), //Min ATR Height of the pole
        LBF( 50 ), // Min distance between flags
        ATRmin( 5 ),// Min volatility change
        K( 1.2 ), //Profit Target constant
        timeexit( 100 ), //Time exit bars
        ATRLL( 3 ),
        BSEMIN( 5 ), // Stop loss below flag
        ATRTRAIL( 3 ),
        TRAILBARS( 5 ), // Trailing stop parameters
        BSEINACT( 70 ),
        ATRINACT( 4 ) ; // Inactivity exit parameter

variables:
        X1( 0 ),
        X2( 0 ),
        LRSX2( 0 ),
        LRSX1( 0 ),
        POLE( 0 ),
        ptarget( 0 ),
        BSE( 0 ),
        TOP( 0 ),
        BOTTOM(  0),
        X3( 0 ),
        L3( 0 ),
        Y23( 0 ),
        FLAGBOT( 0 ),
        UPT1( 0 ),
        LF( 0 ),
        TARGETPER( 0 ),
        MP( 0 ) ;

MP = MarketPosition ;
BSE = BarsSinceEntry ;

if MP <> 0 and MP[1] <> MP then
        BSE = 0
else if MP <> 0 then
        BSE += 1
else if MP = 0 then
        BSE = 0 ;

// FLAG CALCULATION
//Flag duration ex pole top
X1 = HighestBar( Close, MAXFLDUR )[2] ;
//Flag duration including pole top
X2 = X1 + 1 ;
LF = Lowest( Close, X2 ) ;
TOP = Highest( Close, X2 )[2] ;
X2=Iff( LinearRegSlope( Close, X1 )[1] < 0
        and TOP - LF < flagmin * AvgTrueRange( 40 ), X1 + 1, 100 ) ;

if X2 > 2 and X2 <= MAXFLDUR then
        begin //Limits flag duration
        Y23=LowestBar( Close, PX + X2 ) ; // Pole bottom bar
        BOTTOM=Lowest( Close, PX+X2 ) ;// Pole bottom
        POLE = TOP - BOTTOM;
        IF TOP - BOTTOM > POLEMIN * AvgTrueRange( 40 )
                and Y23 > X2 then
                begin
                TOP = Highest( Close, X2 )[2] ;
                FLAGBOT = Lowest( Close, X2 ) ;
                //Uptrend leading to flag
                UPT1 = BOTTOM - Lowest( Low, UPT1BARS ) ;
                //Slope in flag
                LRSX1 = LinearRegSlope( Close, X1 ) * 100 ;
                //Slope in flag before breakout
                LRSX2 = LinearRegSlope( Close, X1-1 )[2] * 100 ;
                //Limits flag slope between 0 and -3 ATR
                Condition1 = TOP-LF < flagmin * AvgTrueRange( 40 )
                        and ( LRSX1 < 0 or LRSX2 < 0 ) ;
                //Limits min pole height
                Condition2 = POLE > POLEMIN * AvgTrueRange( 40 ) ;
                //Uptrend leading to flag
                Condition3 = UPT1 > 0 ;
                //Limits distance between successive flags
                Condition4 = (BarsSinceExit( 1 ) = 0
                        or barssinceexit( 1 ) > LBF ) ;
                //Volatility
                Condition5 = ( AvgTrueRange( 40 ) /
                        AvgTrueRange( 40 )[Y23] - 1 ) * 100 > ATRmin ;
                If MP = 0 and Condition1 and Condition2 and Condition3
                        and Condition4 and Condition5 then
                        begin
                        Buy( "Flag" ) next bar at Highest( Close, X1 ) Stop  ;
                        end ;
                end ;
        end ;

{EXIT CONDITIONS}
if MP = 1 then
        begin
        X3 = HighestBar( Close, MAXFLDUR )[BSE + 2] - BSE + 1 ;
        TOP = Highest( Close, X3 )[BSE + 1] ;
        BOTTOM = Lowest( Close, ( PX + X3 ) )[BSE + 1];
        POLE = ( TOP - BOTTOM ) / ( BOTTOM + .0001 ) * 100 ;
        targetPER = K * POLE ;
        ptarget = ( 1 + TARGETPER / 100 ) * EntryPrice ;
        L3 = Lowest( Low, X3 )[BSE] ;
        //Profit target
        if Close >= ptarget then
                Sell ( "pTARGET" ) this bar on Close;
        //Stop
        if BSE > BSEMIN then
                Sell ( "UNDER FLAG" ) next bar at
                        L3 - ATRLL * AvgTrueRange( 40 ) Stop ;
        //Trailing stop
        if Close < Highest( Close, TRAILBARS )
                - ATRTRAIL * AvgTrueRange( 40 )then
                Sell ( "TRAIL" ) next bar at Market ;
        //Inactivity exit
        if BSE > BSEINACT and Close <
                EntryPrice + ATRINACT * AvgTrueRange( 40 ) then
                Sell ( "INACTIVITY" ) next bar at Market ;
        end ;

//Time exit
if BSE > timeexit then
        Sell ( "TIME" ) next bar at Market ;
```

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

A sample chart is shown in Figure 1.



Symbol CMN | Quantity 500 | Limit Price Auto | Stop Price Auto | Order Type Limit | Route Intelligent | Duration Day+ | Account No. SIM606820M

B:36.94 L:36.93 A:40.54 <Connected> SIMULATED ACCT: Connected to trade server

SA CMN - 15 min NYSE L=36.93 1.46 4.12% B=36.94 A=40.54 O=35.31 Hi=37.06 Lo=35.31 V=108,343 Strategy (15,2.5,23,70,5.5,50,5,1.2,100,3,5,3,5,70,4) Intra...

	Symbol	IntradayFlags
34	CMN	True
35	CO	True
36	CSG	True
37	CSL	True
38	CSLT	True
39	CSV	True
40	CTQ	True
41	CTS	True
42	CVA	True
43	CW	True
44	CWT	True
45	CYD	True
46	CYNI	True
47	DDC	True
48	DEI	True
49	DGX	True
50	DIN	True
51	DOC	True
52	DPD	True
53	DRII	True
54	ELLI	True
55	ENI	True
56	EPR	True
57	ESE	True
58	ESS	True
59	ETJ	True
60	EXL	True
61	EXP	True
62	FCH	True
63	FF	True
64	FNV	True
65	FR	True
66	FRM	True
67	FST	True
68	FT	True
69	GBL	True

Created with TradeStation. ©TradeStation Technologies, Inc. All rights reserved.

FIGURE 1: TRADESTATION, SAMPLE SCANNER RESULTS. Here is a list of some sample results from the TradeStation Scanner along with the indicator and strategy based on Katsanos' article, applied to an intraday chart of Cantel Medical Corp. (CMN).

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

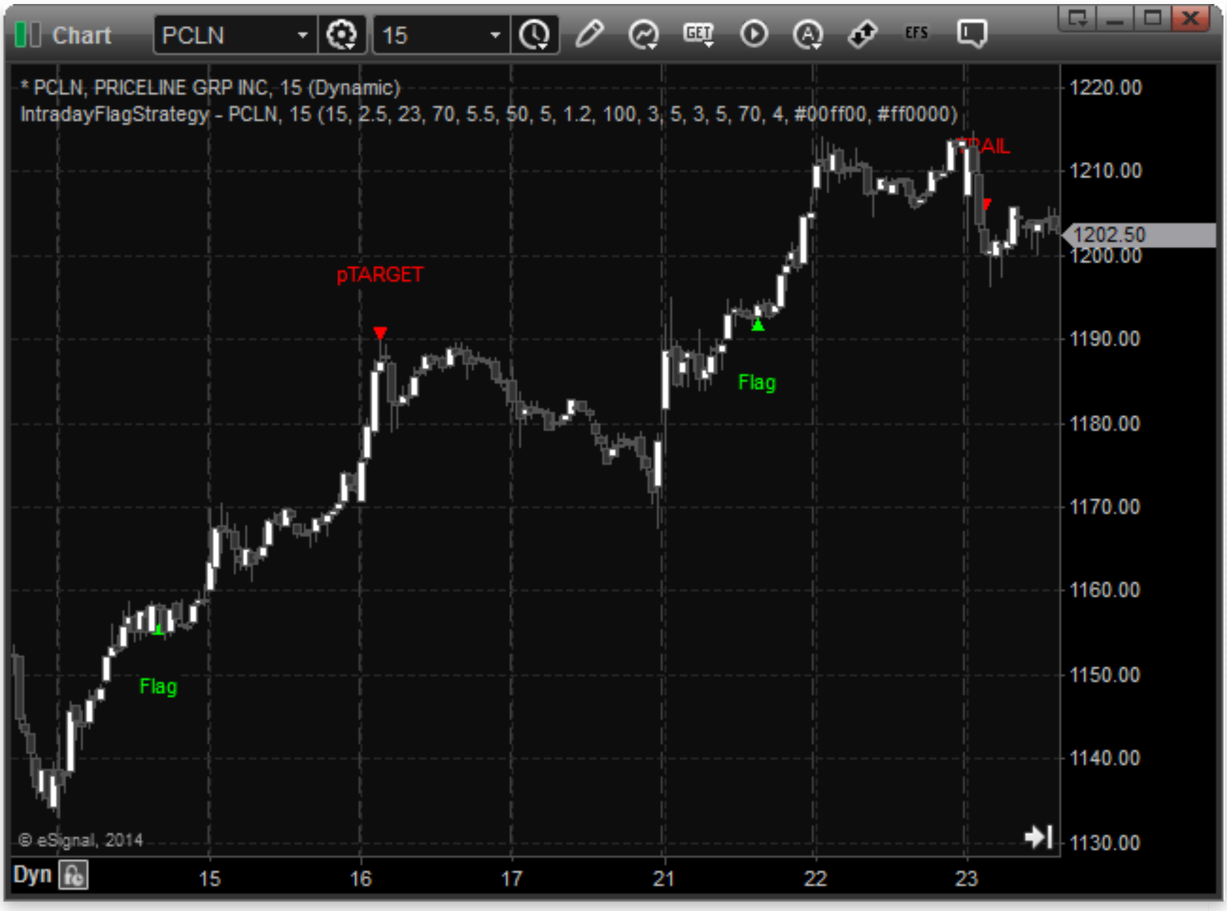
BACK TO LIST

eSignal

eSIGNAL: DECEMBER 2014

For this month's Traders' Tip, we've provided the formula IntradayFlagStrategy.efs based on the formula given in Markos Katsanos' article in this issue, "Detecting Flags In Intraday Charts."

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.



* PCLN, PRICELINE GRP INC, 15 (Dynamic)
IntradayFlagStrategy - PCLN, 15 (15, 2.5, 23, 70, 5.5, 50, 5, 1.2, 100, 3, 5, 3, 5, 70, 4, #00ff00, #ff0000)

FIGURE 2: eSIGNAL. Here is an example of the flag study based on Markos Katsanos' article in this issue implemented on a chart of Priceline Group (PCLN).

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, or downloading here.

```
/**********************************

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:
    Detecting Flags In Intraday Charts by Markos Katsanos

Version:            1.00  10/07/2014

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("IntradayFlagStrategy");
    setPriceStudy(true);

    var x = 0;

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

        setName("Max Flag Duration");
        setDefault(15);
    };


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

        setName("Max ATR in lowest point in flag");
        setDefault(2.5);
    };


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

        setName("Max Pole Duration");
        setDefault(23);
    };


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

        setName("Bars for Uptrend leading to flag");
        setDefault(70);
    };


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

        setName("Min ATR Height of the pole");
        setDefault(5.5);
    };
```

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

    setName("Min distance between flags");
    setDefault(50);
};

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

    setName("Min volatility change");
    setDefault(5);
};

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

    setName("Profit Target constant (K)");
    setDefault(1.2);
};

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

    setName("Time exit bars");
    setDefault(100);
};

fpArray[x] = new FunctionParameter("fpCap1", FunctionParameter.STRING);
```

```
with(fpArray[x++]){

    setName("Stop loss below flag parameters:");
};


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

    setName("    ATRLL");
    setDefault(3);
};

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

    setName("    BSEMIN");
    setDefault(5);
};

fpArray[x] = new FunctionParameter("fpCap2", FunctionParameter.STRING);
with(fpArray[x++]){

    setName("Trailing stop parameters:");
};

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

    setName("   ATRTRAIL");
    setDefault(3);
};

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

    setName("    TRAILBARS");
    setDefault(5);
};

fpArray[x] = new FunctionParameter("fpCap3", FunctionParameter.STRING);
with(fpArray[x++]){

    setName("Inactivity exit parameter:");
};

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

    setName("    BSEINACT");
    setDefault(70);
};

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

    setName("    ATRINACT");
    setDefault(4);
};
```

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

        setName("Enter Position Color");
        setDefault(Color.lime);
    };

    fpArray[x] = new FunctionParameter("fpShortColor", FunctionParameter.COLOR);
    with(fpArray[x++]){

        setName("Exit Position Color");
        setDefault(Color.red);
    };
}

setComputeOnClose();
var bInit = false;
var bVersion = null;

var xClose = null;
var xLow = null;
var xHighestIndexBar = null;
var xTrueRange = null;
var xATR = null;
var xLowestUPT = null;
var xHighestTrail = null;

var BarSinseEntry = null;
var BarSinceExit = 0;
var EntryPrice = null;
var prLinearRegSlopeX2 = null;

function main(fpMaxFlDur, fpFlagMin, fpPX, fpUptBars, fpPoleMin,
              fpDistBFlags, fpATRMin, fpK, fpTimeExit,
              fpATRLL, fpBSEMIN, fpATRTRAIL, fpTRAILBARS,
              fpBSEINACT, fpATRINACT, fpLongColor, fpShortColor){

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

    var nX1 = null;
    var nX2 = null;
    var nOffset = -2;
    var nBottom = null;
    var nPole = null;

    if (getBarState() == BARSTATE_ALLBARS){
        BarSinseEntry = null;
        BarSinceExit = 0;
        EntryPrice = null;
        prLinearRegSlopeX2 = null;
    }

    if (!bInit){

        xClose = close();
        xLow = low();
        xOpen = open();
        xHighestIndexBar = efsInternal("Calc_HighestBar", xClose, fpMaxFlDur);
        xTrueRange = efsInternal("Calc_TrueRange");
        xATR = sma(40, xTrueRange);
        xLowestUPT = lowest(fpUptBars, xLow);
        xHighestTrail = highest(fpTRAILBARS);

        bInit = true;
    };

    var nClose = xClose.getValue(0);
    var nNextOpen = xOpen.getValue(1);
    var nHighestTrail = xHighestTrail.getValue(0);

    nX1 =  xHighestIndexBar.getValue(nOffset);

    if (nX1 == null)
        return;

    nX1 = nX1 + -nOffset

    nX2 =  nX1 + 1;

    var nLF = lowest(nX2).getValue(0);
    var nTop = highest(nX2).getValue(-2);

    var xLinearRegSlopeX1 = efsInternal("Calc_LinearRegression_Slope", xClose, nX1);

    var nPrLinearRegSlope = xLinearRegSlopeX1.getValue(-1);
    var nATR = xATR.getValue(0);
```

```
    if (nLF == null || nTop == null || nPrLinearRegSlope == null || nATR == null ||
nHighestTrail == null)

        return;



    if (nPrLinearRegSlope < 0 && (nTop - nLF) < fpFlagMin * nATR)

        nX2 = nX1 + 1;

    else

        nX2 = 100;



    if (nX2 > 2 && nX2 <= fpMaxFlDur){



        var xLowestIndexBar = efsInternal("Calc_LowestBar", xClose, fpPX + nX2)

        var nY23 = xLowestIndexBar.getValue(0);



        nBottom = lowest(fpPX + nX2).getValue(0);



        if (nY23 == null || nBottom == null)

                return;



        nPole = nTop - nBottom;





        if (nPole > fpPoleMin * nATR && nY23 > nX2){



                nTop = highest(nX2).getValue(-2);

                var nFlagBot = lowest(nX2).getValue(0);

                var nLowestUPT = xLowestUPT.getValue(0);



                var nLinearRegSlopeX1 = xLinearRegSlopeX1.getValue(0);



                var xLinearRegSlopeX2 = efsInternal("Calc_LinearRegression_Slope",
xClose, nX1 - 1);
                var nLinearRegSlopeX2 = xLinearRegSlopeX2.getValue(-2);

                if (nLinearRegSlopeX2 != null)
                    prLinearRegSlopeX2 = nLinearRegSlopeX2;

                if (nLinearRegSlopeX2 == null)
                    nLinearRegSlopeX2 = prLinearRegSlopeX2;

                var nATR_Y23 = xATR.getValue(-nY23)

                if (nTop == null || nLowestUPT == null || nATR_Y23 == null){
                    return;
                }

                var nUPT1 = nBottom - nLowestUPT;

                var nLRSX1 = nLinearRegSlopeX1 * 100;
                var nLRSX2 = nLinearRegSlopeX2 * 100;


                if (getCurrentBarIndex() != 0){

                    if ((!Strategy.isInTrade()) &&
                        (nTop - nLF < fpFlagMin * nATR && (nLRSX1 < 0 || nLRSX2 < 0))
&&
                        (nPole > fpPoleMin * nATR) &&
                        (nUPT1 > 0) &&
                        ( (BarSinceExit == 0) || ( (getCurrentBarCount() -
BarSinceExit) > fpDistBFlags) )&&
                        (nATR / nATR_Y23 - 1) * 100 > fpATRMin) {
                            nPrice = highest(nX1).getValue(0);
                            if (nPrice == null)
                                return;

                            nPrice = Math.max(nNextOpen, nPrice);

                            Strategy.doLong("Flag", Strategy.STOP, Strategy.NEXTBAR,
Strategy.DEFAULT, nPrice)
                            if (Strategy.isInTrade()){
                            drawShapeRelative(1, BelowBar1, Shape.UPTRIANGLE, null,
fpLongColor, Text.PRESET, getCurrentBarCount()+"Flag");
                            drawTextRelative(1, BelowBar2, "Flag", fpLongColor, null,
Text.PRESET|Text.CENTER, null, null, getCurrentBarCount()+"Flag_Text");
                            EntryPrice = nPrice;
                            BarSinseEntry = getCurrentBarCount()+1;
                            return;
                            }

                }
            }
        }
    }

    if (Strategy.isInTrade()){
                var nBSE = getCurrentBarCount() - BarSinseEntry;
                var nX3 = (xHighestIndexBar.getValue(-(nBSE + 2))) - nBSE + 1;
                nX3 = nX3 + nBSE + 2;
                nTop = highest(nX3).getValue(-(nBSE + 1));
                nBottom = lowest(fpPX + nX3).getValue(-(nBSE + 1));
                nPole = (nTop - nBottom)/(nBottom + 0.0001) * 100;
                var nTargetPer = fpK * nPole;
                var nPrTarget = (1 + nTargetPer / 100) * EntryPrice;
                var nL3 = lowest( nX3, low()).getValue(-nBSE);

                if (nClose >= nPrTarget){
                        Strategy.doSell("pTARGET", Strategy.CLOSE, Strategy.THISBAR);
                        drawShapeRelative(0, AboveBar1, Shape.DOWNTRIANGLE, null,
fpShortColor, Text.PRESET, getCurrentBarIndex()+"XLall_t");
                        drawTextRelative(0, AboveBar2, "pTARGET", fpShortColor, null,
Text.PRESET|Text.CENTER, null, null, getCurrentBarIndex()+"XLall_ts");
                        BarSinceExit = getCurrentBarCount();
                        return;
                }

                if (nBSE > fpBSEMIN){
                        var nSell = nL3 - fpATRLL * nATR;

                        nSell = Math.min(nNextOpen, nSell);

                        Strategy.doSell("UNDER FLAG", Strategy.STOP,
Strategy.NEXTBAR, Strategy.DEFAULT, nSell)
                        if (!Strategy.isInTrade()){
                        drawShapeRelative(1, AboveBar1, Shape.DOWNTRIANGLE, null,
fpShortColor, Text.PRESET, getCurrentBarIndex()+"XLall_u");
                        drawTextRelative(1, AboveBar2, "UNDER FLAG", fpShortColor,
null, Text.PRESET|Text.CENTER, null, null, getCurrentBarIndex()+"XLall_ut");
                        BarSinceExit = getCurrentBarCount();
                        return;
                        }
                }

                if (nClose < nHighestTrail - fpATRTRAIL * nATR){
                        Strategy.doSell("TRAIL", Strategy.MARKET, Strategy.NEXTBAR)
                        drawShapeRelative(1, AboveBar1, Shape.DOWNTRIANGLE, null,
fpShortColor, Text.PRESET, getCurrentBarIndex()+"XLall_tr");

                        drawTextRelative(1, AboveBar2, "TRAIL", fpShortColor, null,
Text.PRESET|Text.CENTER, null, null, getCurrentBarIndex()+"XLall_trt");

                        BarSinceExit = getCurrentBarCount() + 1;

                        return;

                }



                if (nBSE > fpBSEINACT && nClose < EntryPrice + fpATRINACT*nATR){

                        Strategy.doSell("INACTIVITY", Strategy.MARKET,
Strategy.NEXTBAR)

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

                        drawTextRelative(1, AboveBar2, "INACTIVITY", fpShortColor,
null, Text.PRESET|Text.CENTER, null, null, getCurrentBarIndex()+"XLall_it");

                        BarSinceExit = getCurrentBarCount();

                        return;

                }



                if (nBSE > fpTimeExit){

                        Strategy.doSell("TIME", Strategy.MARKET, Strategy.NEXTBAR)

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

                        drawTextRelative(1, AboveBar2, "TIME", fpShortColor, null,
Text.PRESET|Text.CENTER, null, null, getCurrentBarIndex()+"XLall_mt");

                        BarSinceExit = getCurrentBarCount();

                        return;

                }



    }

    return;

}
```

```
function Calc_HighestBar(xSource, nLength){

    var nBarNumber = 0;
    var nBarValue = xSource.getValue(0);

    for (i = 1; i <= nLength - 1; i++){
        var nSource = xSource.getValue(-i);
        if (nSource == null)
            return;
        if (nSource > nBarValue){
                nBarValue = nSource;
                nBarNumber = i;
        }
    }

    return nBarNumber;
}

function Calc_LowestBar(xSource, nLength){

    var nBarNumber = 0;
    var nBarValue = xSource.getValue(0);

    for (i = 1; i <= nLength - 1; i++){
        var nSource = xSource.getValue(-i);
        if (nSource == null)
            return;
        if (nSource < nBarValue){
                nBarValue = nSource;
                nBarNumber = i;
        }
    }

    return nBarNumber;
}



function Calc_LinearRegression_Slope(xSourse, nPeriod)
{
    if (getCurrentBarCount() < nPeriod)
        return;

    var nPer = 1;
    var nTime = 0;
    var nTSq = 0;
    var nPrice = 0;
    var nTimePrice = 0;
    var nVal = 0;
    var nSlope = 0;

    for (var i = 0; i < nPeriod; i++)
    {
       nTime += nPer;
       nVal = xSourse.getValue(-(nPeriod - 1) + i);
       nPrice += nVal;
       nTimePrice += (nVal * nPer);
       nTSq += (nPer * nPer);
       nPer++;
    }
    nPer--;


    nSlope = ((nPer * nTimePrice) - (nTime * nPrice)) / ((nPer * nTSq) - (nTime *
nTime));


    return nSlope;

}



var xClose = null;

var xHigh = null;

var xLow = null;



function Calc_TrueRange(){


    if (getBarState() == BARSTATE_ALLBARS)

    {

        xClose = close();

        xHigh = high();

        xLow = low();

    }


    var nPrClose = xClose.getValue(-1);


    if (nPrClose == null)

        return;


    var nHigh = xHigh.getValue(0);

    var nLow = xLow.getValue(0);


    var nTrueHigh = null;
    var nTrueLow = null;


    if (nPrClose > nHigh)

        nTrueHigh = nPrClose

    else

        nTrueHigh = nHigh;


    if (nPrClose < nLow)

        nTrueLow = nPrClose

    else

        nTrueLow = nLow;


    var nReturnValue = nTrueHigh - nTrueLow;


    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

THINKORSWIM: DECEMBER 2014

Just in time for the holidays, technician Markos Katsanos has delivered us a fantastic gift in the form of his article in this issue, "Detecting Flags In Intraday Charts." In the article, Katsanos demonstrates that even the most classic of flag patterns can be used with intraday data.

At thinkorswim, we've used our proprietary scripting language *thinkScript* to build a strategy for detecting trends using this method. We've made the loading process extremely easy — simply click on the link http://tos.mx/TQSAVh and choose *Backtest in thinkorswim*, then choose to rename your study to "IntradayFlagFormation." You can adjust the parameters of these within the *edit studies* window to fine-tune your variables.

You can see from the chart in Figure 3 that the strategy on thinkorswim charts will give you entry points — shown as the two blue arrows — when a flag is beginning and when the point forms. You will also see that the peak of the pole is the indicator with the exit point. To see how this Strategy performed simply right click on the exit point and choose Show report. For a detailed description see the article in Technical Analysis of STOCKS & COMMODITIES. Happy swimming!

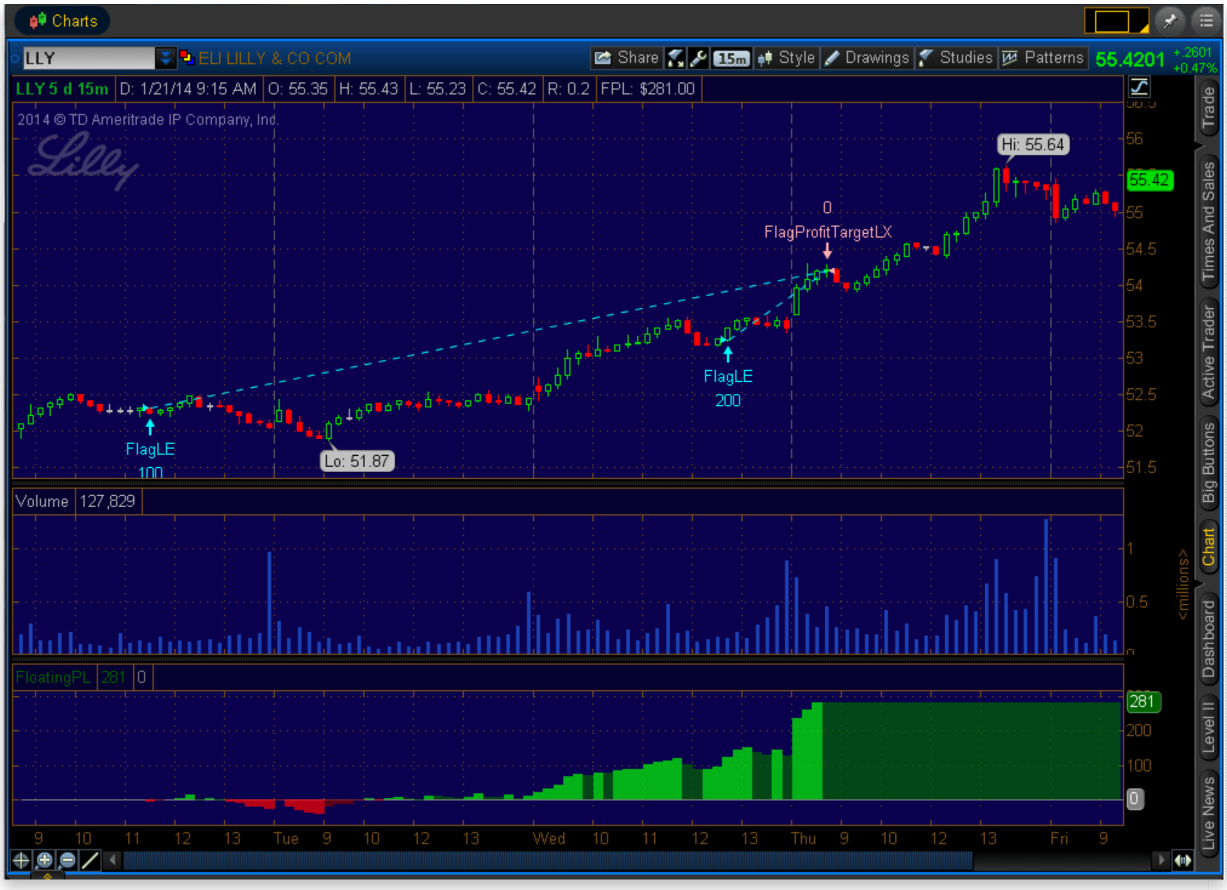


FIGURE 3: THINKORSWIM. The two blue arrows show entry points for the strategy when a flag is beginning and when the point forms. The peak of the pole is the indicator with the exit point.

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

BACK TO LIST

WEALTH-LAB: DECEMBER 2014

Detecting chart patterns is always a special joy to code if the provided rules are well-thought-out and fully mechanical. Such is the clear definition of flag patterns given by author Markos Katsanos in his article in this issue, “Detecting Flags In Intraday Charts.”

We’ve used the following simplified set of rules to detect flags:

- A steep pole of A times the average true range (ATR) in B bars
- A flag breaking out in C bars or less from the pole top and sloping horizontally

- Flag depth not more than D times the ATR measured from the highest to the lowest point in the flag
- An uptrend during the last E bars leading up to the pole.

See Figure 4 for a sample chart.

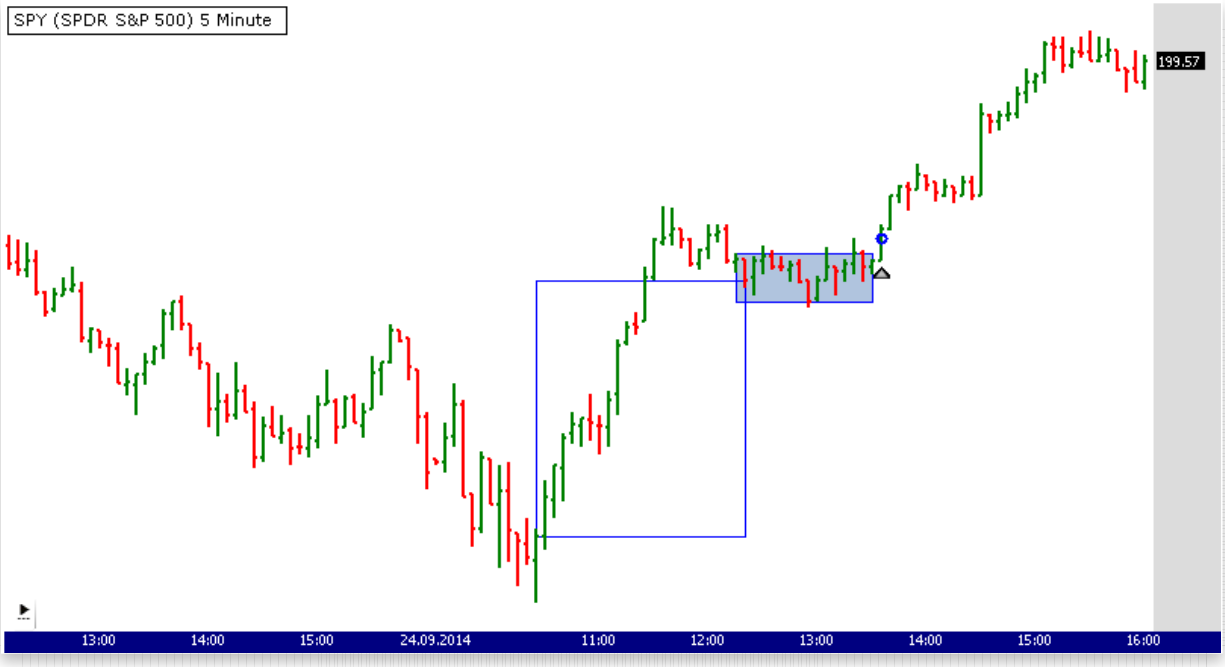


FIGURE 4: WEALTH-LAB, FLAG PATTERN. This sample Wealth-Lab 6 chart illustrates the detection of the flag pattern on a five-minute chart of the SPY (S&P 500 SPDR).

With minimal tweaks to the system's parameters, this set of rules can be applied to charts of different time frames (such as end-of-day, Figure 5).

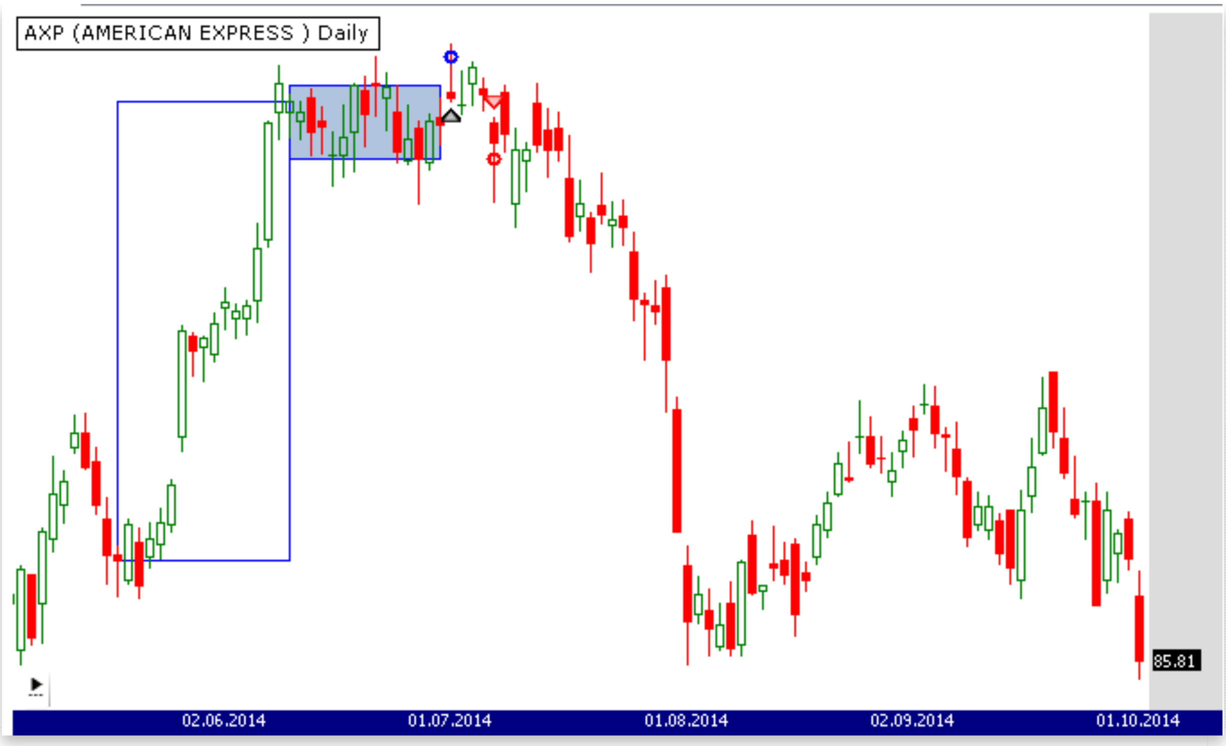


FIGURE 5: WEALTH-LAB, FAILED FLAG. This shows a failed flag on the daily chart of American Express (AXP) in Wealth-Lab 6.

Yet there's room for improvement: Consider adding a filter against erratic price movement; phase out the less probable trades against the medium-term (daily) trend; and play around with the exits. (To our taste, the initial stop at the flag bottom may result in premature exits; for example, subtracting an ATR from that level could be a more robust approach.)

In addition, we already offer a similar system that looks for tight consolidation ranges, and it is available to Wealth-Lab users along with other related systems that mechanically identify chart patterns. To download it, first download all publicly available strategies (that is, click *download* in the *open strategy* dialog). Then look for the strategy named "Rectangle Trading System (Acme R)" in the *chart patterns* folder.

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

namespace WealthLab.Strategies
{
        public class Katsanos201412 : WealthScript
        {
                private StrategyParameter paramPoleTimeout;
                private StrategyParameter paramFlagTimeout;
                private StrategyParameter paramUptrendBeforePole;
                private StrategyParameter paramMinFlagDuration;
                private StrategyParameter paramMaxFlagDuration;
                private StrategyParameter paramSMAPeriod;
                private StrategyParameter paramInactivityStop;
                private StrategyParameter paramTimeout;

                private StrategyParameter paramPoleHeight;
                private StrategyParameter paramFlagHeight;
                private StrategyParameter paramProfitTarget;

                void DrawRectangle(int b1, int b2, double p1, double p2, Color c)
                {
                        double[] rect = { b1, p1, b1, p2, b2, p2, b2, p1 };
                        DrawPolygon( PricePane, Color.Blue, c, LineStyle.Solid, 1,
true, rect );
                }

                public Katsanos201412()
                {
                        paramPoleTimeout = CreateParameter("Pole Timeout", 23, 10,
50, 1);
                        paramPoleHeight = CreateParameter("Pole Height", 5.5, 1.0,
10, 0.5);
                        paramUptrendBeforePole = CreateParameter("Uptrend Before
Pole", 70, 10, 100, 10);
                        paramSMAPeriod = CreateParameter("SMA Period", 50, 10, 100,
5);

                        paramFlagHeight = CreateParameter("Flag Height", 2.5, 0.5,
5.0, 0.5);
                        paramFlagTimeout = CreateParameter("Flag Timeout", 15, 3, 30,
1);
                        paramMinFlagDuration = CreateParameter("Min Flag Duration",
3, 3, 30, 1);

                        paramInactivityStop = CreateParameter("Inactivity Stop", 70,
10, 100, 10);
                        paramTimeout = CreateParameter("Timeout", 100, 10, 100, 10);
                        paramProfitTarget = CreateParameter("Profit Target ATR", 1.2,
0.2, 3.0, 0.2);
                }

                protected override void Execute()
                {
                        int PoleTimeout = paramPoleTimeout.ValueInt,
                                FlagTimeout = paramFlagTimeout.ValueInt,
                                UptrendLeadingToPole =
paramUptrendBeforePole.ValueInt,
                                MinFlagDuration = paramMinFlagDuration.ValueInt,
                                smaPeriod = paramSMAPeriod.ValueInt,
                                inactivityStop = paramInactivityStop.ValueInt,
                                timeout = paramTimeout.ValueInt,
                                PoleBar = 0, FlagBar = 0, ba = 0;

                        double poleHeight = paramPoleHeight.Value,
                                flagHeight = paramFlagHeight.Value,
                                currPoleHeight = 0,
                                ProfitTarget = paramProfitTarget.Value,
                                InitialStop = 0, ws =  0.5, flagTop = 0, flagBottom =
0;
                        bool PoleValid = false, FlagValid = false;
```

```
                        SMA sma = SMA.Series( Close,smaPeriod );
                        LinearRegSlope lrs = LinearRegSlope.Series( Close,
FlagTimeout );
                        HideVolume();

                        for(int bar = GetTradingLoopStartBar(100); bar < Bars.Count;
bar++)
                        {
                                if (IsLastPositionActive)
                                {
                                        // Exits
                                        Position p = LastPosition;
                                        double atr = ATR.Series( Bars, 40 )[bar];
                                        double high = p.HighestHighAsOfBar(bar);
                                        double chandelier = high - atr * 3;
                                        double inactivity = atr * 4;

                                        if( ( bar+1 - p.EntryBar >= inactivityStop )
&& ( p.MFEAsOfBar( bar ) < inactivity ) )
                                                SellAtMarket( bar+1, p,
"Inactivity+MFE" );
                                                else
                                                if( bar+1 - p.EntryBar >= timeout )
                                                        SellAtMarket( bar+1, p, "Time
exit" );
                                        else
                                        if( !SellAtStop( bar+1, p, p.RiskStopLevel,
"Stop loss" ) )
                                                if( !SellAtStop( bar+1, p,
chandelier, "Trailing (Chandelier)" ) )
                                                        SellAtLimit( bar+1, p,
p.AutoProfitLevel, "Profit Target" );
                                }
                                else
                                {

                                        if( !PoleValid )
                                        {
                                                //Uptrend during the last 70 bars
leading to the pole.

                                                if(Lowest.Value( bar, Close,
PoleTimeout ) > Lowest.Value( bar, Close, UptrendLeadingToPole ))
                                                {
                                                        //A steep pole of 5.5 times
the average true range (ATR) or more, in 23 bars or less.

                                                        currPoleHeight =
Close[PoleBar] - Close[bar - PoleTimeout];
                                                        double atr = ATR.Value(bar,
Bars, 40);
                                                        PoleBar = bar;
                                                        PoleValid = currPoleHeight >=
atr * poleHeight ? true: false;
                                                }
                                        }

                                        if( PoleValid )
                                        {
                                                if( !FlagValid )
                                                {
                                                        //A flag breaking out in 15
bars or less from the pole top and sloping horizontally or slightly down.

                                                        if( bar <= PoleBar +
FlagTimeout && bar >= PoleBar + MinFlagDuration ) // To avoid premature triggering
                                                        {
                                                                flagTop =
Highest.Value( bar, Close, FlagTimeout );
                                                                flagBottom =
Lowest.Value( bar, Close, FlagTimeout );
                                                                InitialStop =
flagBottom;
                                                                double flagRange =
flagTop - flagBottom;
                                                                double atr =
ATR.Value(bar, Bars, 40);
                                                                double slope =
lrs[bar];
                                                                bool isSlopeOK =
slope > -0.04 && slope <= 0.01;

                                                                //Flag depth not more
than 2.5 times the ATR measured from the highest to the lowest point in the flag.

                                                                if( flagRange <= atr
* flagHeight && isSlopeOK )
                                                                {
                                                                        FlagValid =
true; FlagBar = bar;
                                                                }
                                                        }
                                                        else
                                                                PoleValid = bar + 1 -
PoleBar < PoleTimeout;  // reset if Setup has timed out
                                                }

                                                if( FlagValid )
                                                {
                                                        if( BuyAtStop( bar + 1,
Highest.Value(bar, High, FlagTimeout) ) != null )
                                                        {
                                                                // Draw flag and pole
                                                                DrawRectangle(
FlagBar, FlagBar-FlagTimeout, flagTop, flagBottom, Color.LightSteelBlue );
                                                                DrawRectangle(
PoleBar, PoleBar-PoleTimeout, Close[PoleBar], Close[PoleBar-PoleTimeout],
Color.Transparent );

                                                                // Assign initial
stop and profit target levels

LastPosition.AutoProfitLevel = LastPosition.EntryPrice + currPoleHeight *
ProfitTarget;

LastPosition.RiskStopLevel = InitialStop;

                                                                PoleValid = false;
FlagValid = false;      // reset Setup variables
```

```
                                                        }
                                                        else
                                                                // reset if Setup has
timed out

                                                        {
                                                                PoleValid = bar + 1 -

PoleBar < PoleTimeout;

                                                                FlagValid = false;
                                                                flagTop = 0;
                                                                flagBottom = 0;
                                                        }
                                                }
                                        }
                                }
                        }
                }
        }
}
```

To sum up, the proposed technique is useful to recognize brief retreats in steep trends.

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

BACK TO LIST

NEUROSHELL TRADER: DECEMBER 2014

The strategy presented in Markos Katsanos' article in this issue for flag detection ("Detecting Flags In Intraday Charts") can be easily implemented in NeuroShell Trader using NeuroShell Trader's ability to call external programs. The programs may be written in C, C++, Power Basic, or Delphi.

After moving the EasyLanguage code given in Katsanos' article to your preferred compiler and creating a dynamic link library (DLL), you can insert the resulting entry and profit target 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. Setup the parameters to match your DLL.
5. Select the *finished* button.

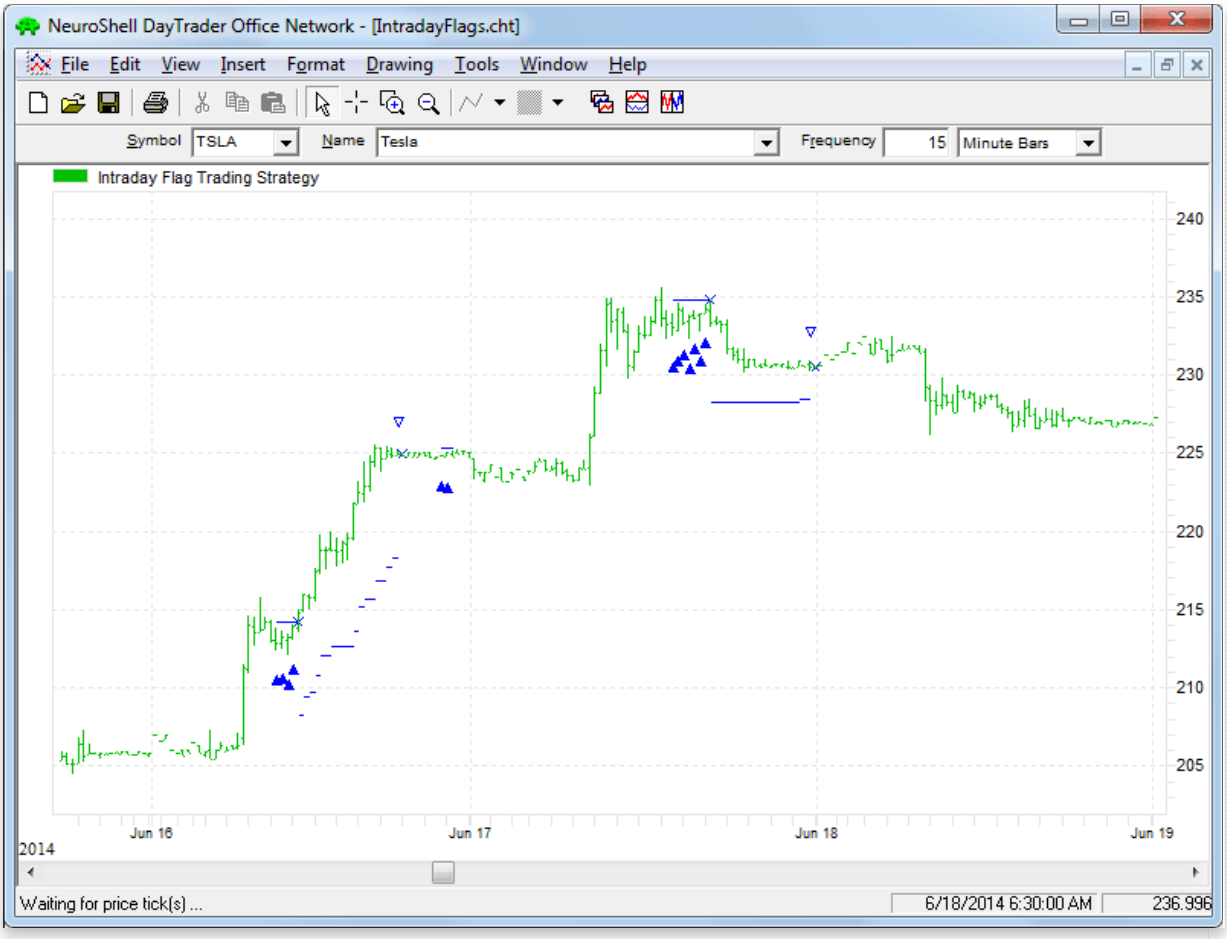
To recreate the intraday flag trading system, select "New trading strategy" from the Insert menu and enter the following in the appropriate locations of the trading strategy wizard:

```
BUY LONG CONDITIONS:
        Intraday Flag Entry Signal( High, Low, Close)

BUY STOP PRICE:
        Intraday Flag Entry Stop( High, Low, Close)

LONG TRAILING STOP PRICES:
        TrailPriceATR(Trading Strategy, 40, 3)

SELL LONG CONDITIONS: [1 of which must be true]
        Intraday Flag Profit Target Signal( High, Low, Close)
        Inactivity%(Trading Strategy, 10, 15)
```

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 website to download a copy of this or any previous Traders' Tips.

A sample chart is shown in Figure 6.


NeuroShell DayTrader Office Network - [IntradayFlags.cht]

Symbol TSLA Name Tesla Frequency 15 Minute Bars

Intraday Flag Trading Strategy

Waiting for price tick(s) ...

6/18/2014 6:30:00 AM 236.996

FIGURE 6: NEUROSHELL TRADER, intraday flag

strategy. This NeuroShell Trader chart displays the intraday flag trading strategy.

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

BACK TO LIST

AMIBROKER: DECEMBER 2014

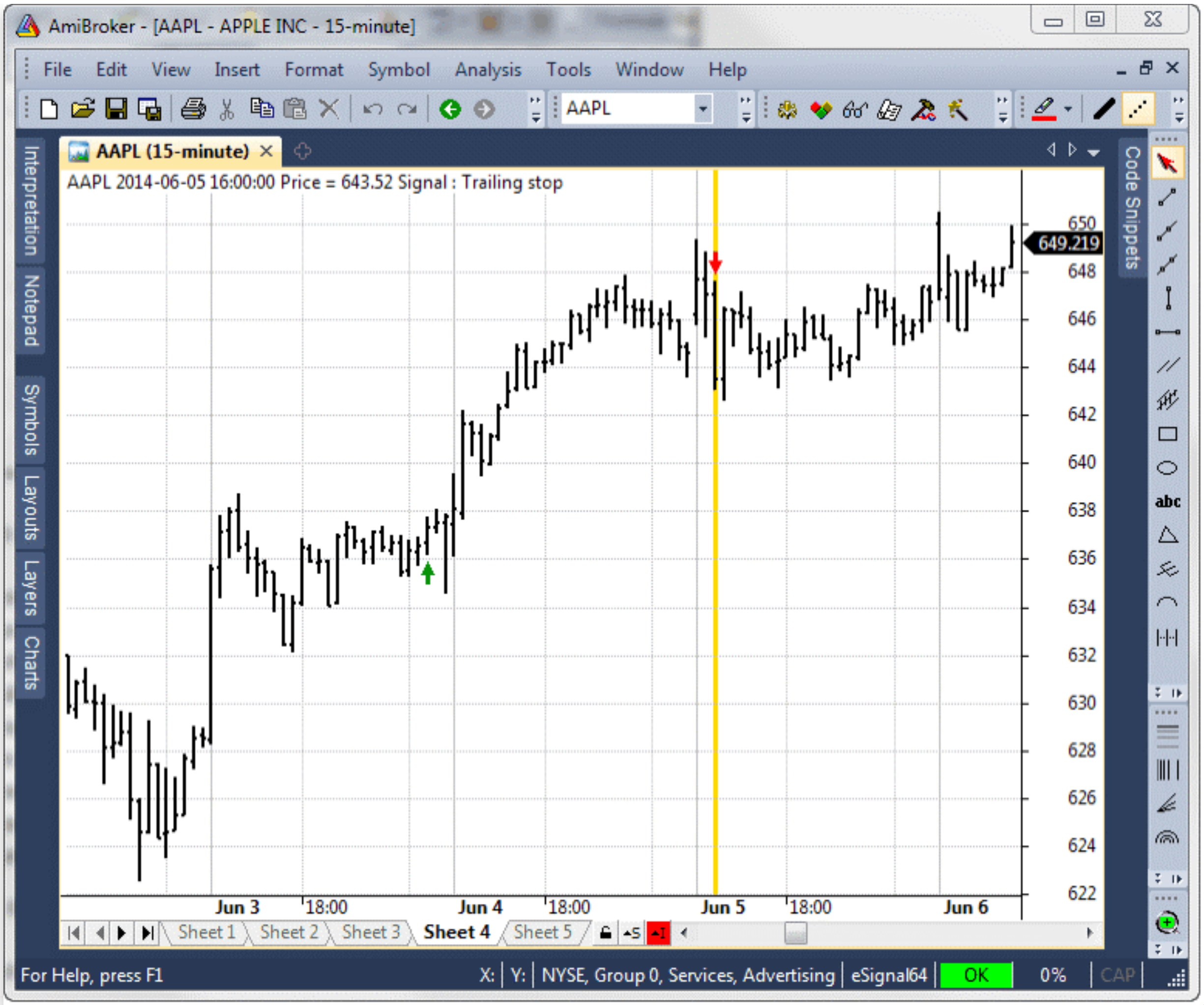
In “Detecting Flags In Intraday Charts” in this issue, author Markos Katsanos proposes using flag chart formations as a entry signal for 15-minute chart trading. Chart formations are highly subjective and difficult to express in strict mathematical rules. What is “clearly visible” to the eye is not that clear when mathematical rules are applied. In the article, Katsanos has made an attempt to code flag formation detection and provides EasyLanguage code for his technique. The formula we are presenting is more or less a direct translation of his code into AmiBroker Formula Language (AFL).

The AFL code is shown below, as well as at the AmiBroker site.

```
LISTING 1.
maxfldur = 15; // max flag duration
flagmin = 2.5; // max atr in lowest point in flag
px = 23; //Max Pole Duration.
upt1bars = 70; // Bars for Uptrend leading to flag
polemin = 5.5; //Min ATR Height of the pole
lbf = 50; // Min distance between flags
ATRmin = 5;// Min volatility change
K = 1.2; //Profit Target constant
timeexit = 100; //Time exit bars
atrLL = 3; // max loss distance in ATR units
atrTrail = 3; // trail distance in ATR uints

// FLAG CALCULATION
X1 = Ref( HHVBars( C, maxfldur ), -2 );//Flag duration ex pole top
X2 = X1 + 1; //Flag duration including pole top
LF = LLV( C, X2 );
TOP = Ref( HHV( C, X2 ), -2 );
X2 = IIF( Ref( LinRegSlope( C, X1 ), -1 ) < 0 AND TOP - LF < flagmin * ATR( 40 ), X1
+ 1, 100 );

PreCond1 = X2 > 2 AND X2 <= maxfldur; //Limits flag duration
Y23 = LLVBars( C, PX + X2 ); // Pole bottom bar
BOTTOM = LLV( C, ( PX + X2 ) );// Pole bottom
POLE = TOP - BOTTOM;
PreCond2 = TOP - BOTTOM > polemin * ATR( 40 ) AND Y23 > X2;
TOP = Ref( HHV( C, X2 ), -2 );
```

```
FLAGBOT = LLV( C, X2 );
UPT1 = BOTTOM - LLV( L, upt1bars ); //Uptrend leading to flag
LRSX1 = LinRegSlope( C, X1 ) * 100; //Slope in flag
LRSX2 = Ref( LinRegSlope( C, X1 - 1 ), -2 ) * 100; //Slope in flag before breakout
Condition1 = TOP - LF < flagmin * ATR( 40 ) AND ( LRSX1 < 0 OR LRSX2 < 0 ); //Limits
flag slope between 0 and -3 ATR
Condition2 = POLE > polemin * ATR( 40 ); //Limits min pole height
Condition3 = UPT1 > 0; //Uptrend leading to flag
Condition4 = ( ATR( 40 ) / Ref( ATR( 40 ), Y23 ) - 1 ) * 100 > ATRmin; //Volatility

TriggerPrice = HHV( C, X1 );

Buy = PreCond1 AND PreCond2 AND Condition1 AND Condition2 AND Condition3 and
Condition4 AND H > TriggerPrice;
BuyPrice = TriggerPrice;

Sell = 0; // all exits using stops

ApplyStop( stopTypeNBar, stopModeBars, timeexit ); // time exit
ApplyStop( stopTypeProfit, stopModePoint, k * pole, True ); // profit target
ApplyStop( stopTypeTrailing, stopModePoint, atrTrail * ATR( 40 ), True, True ); //
trailing
ApplyStop( stopTypeLoss, stopModePoint, atrLL * ATR( 40 ), True );

Equity( 1 ); // evaluate stops

Plot( C, "Price", colorDefault, styleBar | styleThick );
PlotShapes( IIf( Buy, shapeUpArrow, 0 ), colorGreen, 0, Low );
PlotShapes( IIf( Sell, shapeDownArrow, 0 ), colorRed, 0, High );

Title = Name() + " " + Date() + " Price = " + Close + " Signal : " + WriteIf( Buy,
"Entry",
         WriteIf( Sell == 2, "Max loss",
         WriteIf( Sell == 3, "Profit target",
         WriteIf( Sell == 4, "Trailing stop",
         WriteIf( Sell == 5, "Time stop", "None" ) ) ) ) );
```

A sample chart demonstrating flag detection based on Katsanos’ approach is shown in Figure 7.



AmiBroker - [AAPL - APPLE INC - 15-minute]

AAPL 2014-06-05 16:00:00 Price = 643.52 Signal : Trailing stop

For Help, press F1

FIGURE 7: AMIBROKER, FLAG. Here is an example of a detected flag formation on a 15-minute chart of AAPL. The buy signal is represented by the green arrow and a trailing stop is represented by the red arrow.

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

BACK TO LIST

NINJATRADER: DECEMBER 2014

The intraday flag strategy presented in “Detecting Flags In Intraday Charts” by Markos Katsanos in this issue has been made available for download at www.ninjatrader.com/SC/December2014SC.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 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 "IntradayFlags" file.

A sample chart implementing the strategy is shown in Figure 8.

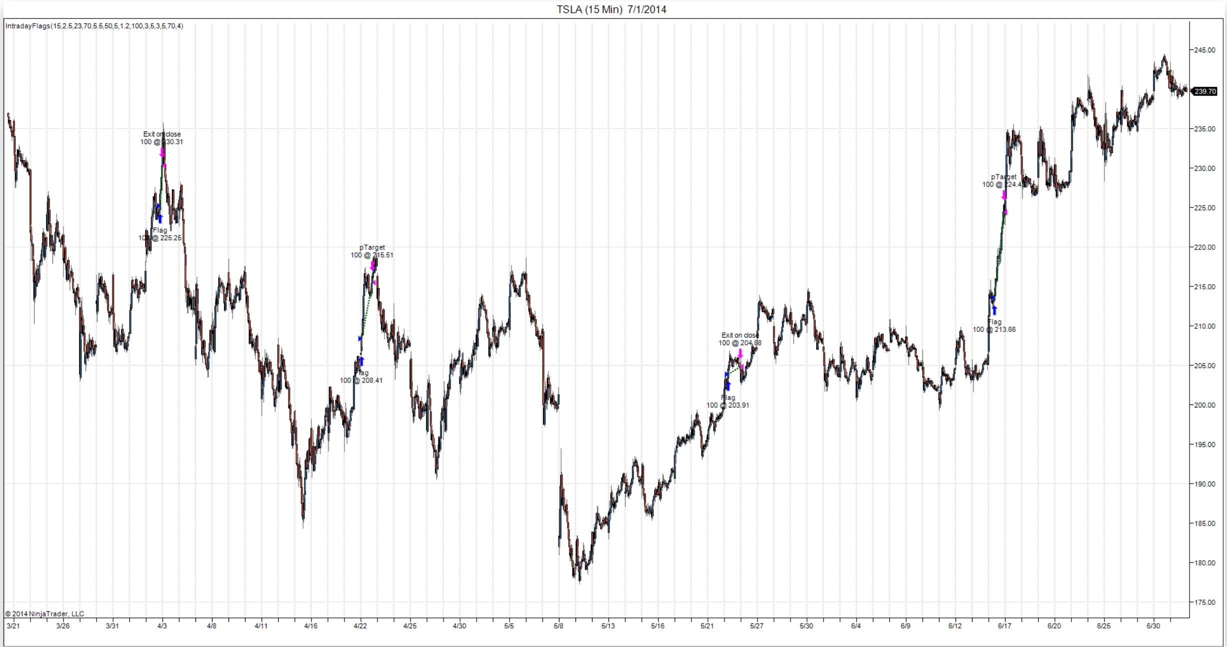


FIGURE 8: NINJATRADER. The screenshot shows the IntradayFlags strategy applied to a 15-minute chart of Tesla Motors (TSLA) in NinjaTrader.

—Raymond Deux & Paul Hunt
NinjaTrader, LLC
www.ninjatrader.com

BACK TO LIST

UPDATA: DECEMBER 2014

Our Traders' Tip for this month is based on "Detecting Flags In Intraday Charts" by Markos Katsanos in this issue.

In the article, the author proposes a set of rules to find flag continuation patterns in intraday financial data. Flags are defined according to five key criteria or conditions, mostly relating to some minimum height and width requirements of the flag, as measured by average true range (ATR). Once you are in a trade, there are five types of time-based or price-based exit strategies you can use.

We have added some custom Updata code based on Katsanos' article to the Updata library, which may be downloaded by clicking the *custom* menu and then *system library*. Those who cannot access the library due to a firewall may copy the code shown below and paste it into the Updata custom editor and save it. See Figure 9 for a sample chart.

```
'Intraday Flag Strategy
PARAMETER "Max Flag Duration" #MAXFLDUR=15
PARAMETER "Max ATR @ Flag Low" @FLAGMIN=2.5
PARAMETER "Max Pole Duration" #PX=23
PARAMETER "Bars Up To Flag" #UPTIBARS=70
PARAMETER "Min ATR Pole Height" @POLEMIN=5.5
PARAMETER "Min Flag Distance" #LBF=50
PARAMETER "Min Volatility Change" #ATRMin=5
PARAMETER "Profit Target Const." @K=1.2
PARAMETER "Time Exit Bars" #TIMEEXIT=100
PARAMETER "Below Flag StopLoss" #ATRLL=3
PARAMETER "Trailing Stop ATR" @ATRTRAIL=3
PARAMETER "Trailing Stop Bars" #TRAILBARS=5
PARAMETER "Entry Inactivity Bars" #BSEINACT=70
PARAMETER "Entry Inactivity ATR" #ATRINACT=4

@X1=0
@X2=0
@X3=0
@LF=0
@TOP=0
@Y23=0
@BOTTOM=0
@POLE=0
@FLAGBOT=0
@UPT1=0
@LRSX1=0
@LRSX2=0
@CONDITION1=0
@CONDITION2=0
@CONDITION3=0
@CONDITION4=0
@CONDITION5=0
#BARSSINCEEXIT=0
#BUYORDER=0
@BUYENTRYLEVEL=0
@ENTRYPRICE=0
@targetPER=0
@targetPRICE=0
@L3=0
@TRAILINGSTOP=100000
@STOP=0
FOR #CURDATE=0 TO #LASTDATE
   'FLAG CALCULATION
   @X1=PHIGH(CLOSE,#MAXFLDUR)
   @X2=@X1+1
```

```
@LF=PLOW(CLOSE,@X2)
@TOP=PHIGH(CLOSE,@X2)
IF (LSR(CLOSE,@X2,0,0)-LSR(CLOSE,@X2,0,@X2))<0 AND (@TOP-@LF)<@FLAGMIN*ATR(40)
   @X2=@X1+1
ELSE
  @X2=100
ENDIF
IF @X2>2 AND @X2<#MAXFLDUR
   @Y23=PLOW(CLOSE,#PX+@X2)
   @BOTTOM=PLOW(CLOSE,#PX+@X2)
   @POLE=@TOP-@BOTTOM
ENDIF
IF @POLE>@POLEMIN*ATR(40) AND @Y23>@X2
   @TOP=PHIGH(CLOSE,@X2)
   @FLAGBOT=PLOW(CLOSE,@X2)
   @UPT1=@BOTTOM-PLOW(LOW,#UPTIBARS)
ENDIF
@LRSX1=(LSR(CLOSE,@X1,0,0)-LSR(CLOSE,@X1,0,@X1))*(100/@X1)
@LRSX2=(LSR(CLOSE,@X1-1,0,0)-LSR(CLOSE,@X1-1,0,@X1-1))*(100/(@X1-1))
@CONDITION1=(@TOP-@LF)<@FLAGMIN*ATR(40) AND (@LRSX1<0 OR @LRSX2<0)
@CONDITION2=@POLE>@POLEMIN*ATR(40)
@CONDITION3=@UPT1>0
@CONDITION4=(#BARSSINCEEXIT=0) OR (#BARSSINCEEXIT>#LBF)
@CONDITION5=ATR(40)/HIST(ATR(40),@Y23-1)*100>#ATRMin
IF #BUYORDER=TRUE AND HIGH>@BUYENTRYLEVEL
   BUY @BUYENTRYLEVEL
   #BUYORDER=FALSE
   @ENTRYPRICE=@BUYENTRYLEVEL
ENDIF
IF @CONDITION1=@CONDITION2=@CONDITION3=@CONDITION4=@CONDITION5=1
   #BUYORDER=TRUE
   @BUYENTRYLEVEL=PHIGH(CLOSE,@X1)
ENDIF
'EXIT CONDITIONS
IF ORDERISOPEN=1
   @X3=PHIGH(CLOSE,ORDEROPENFOR+#MAXFLDUR)
   @TOP=PHIGH(CLOSE,@X3+ORDEROPENFOR)
   @BOTTOM=PLOW(CLOSE,#PX+@X3)
   @POLE=100*(@TOP-@BOTTOM)/(@BOTTOM+0.0001)
   @targetPER=@K*@POLE
   @targetPRICE=(1+@targetPER/100)*@ENTRYPRICE
   @L3=PLOW(LOW,@X3+ORDEROPENFOR)
ENDIF
'PROFIT TARGET
IF CLOSE>@targetPRICE
   SELL @targetPRICE
ENDIF
'STOP
IF ORDEROPENFOR>#BSEINACT
   @STOP=@L3-#ATRLL*ATR(40)
   If CLOSE<@STOP
      SELL @STOP
    EndIf
ENDIF
'TRAILING STOP
IF HIST(CLOSE<PHIGH(CLOSE,#TRAILBARS)-@ATRTRAIL*ATR(40),1)
   SELL CLOSE
ENDIF
'INACTIVITY EXIT
IF ORDEROPENFOR>#BSEINACT AND HIST(CLOSE<@ENTRYPRICE+#ATRINACT*ATR(40),1)
```

```
      SELL CLOSE
   ENDIF
   'TIME EXIT
   IF HIST(ORDEROPENFOR>#TIMEEXIT,1)
      SELL CLOSE
   ENDIF
NEXT
```

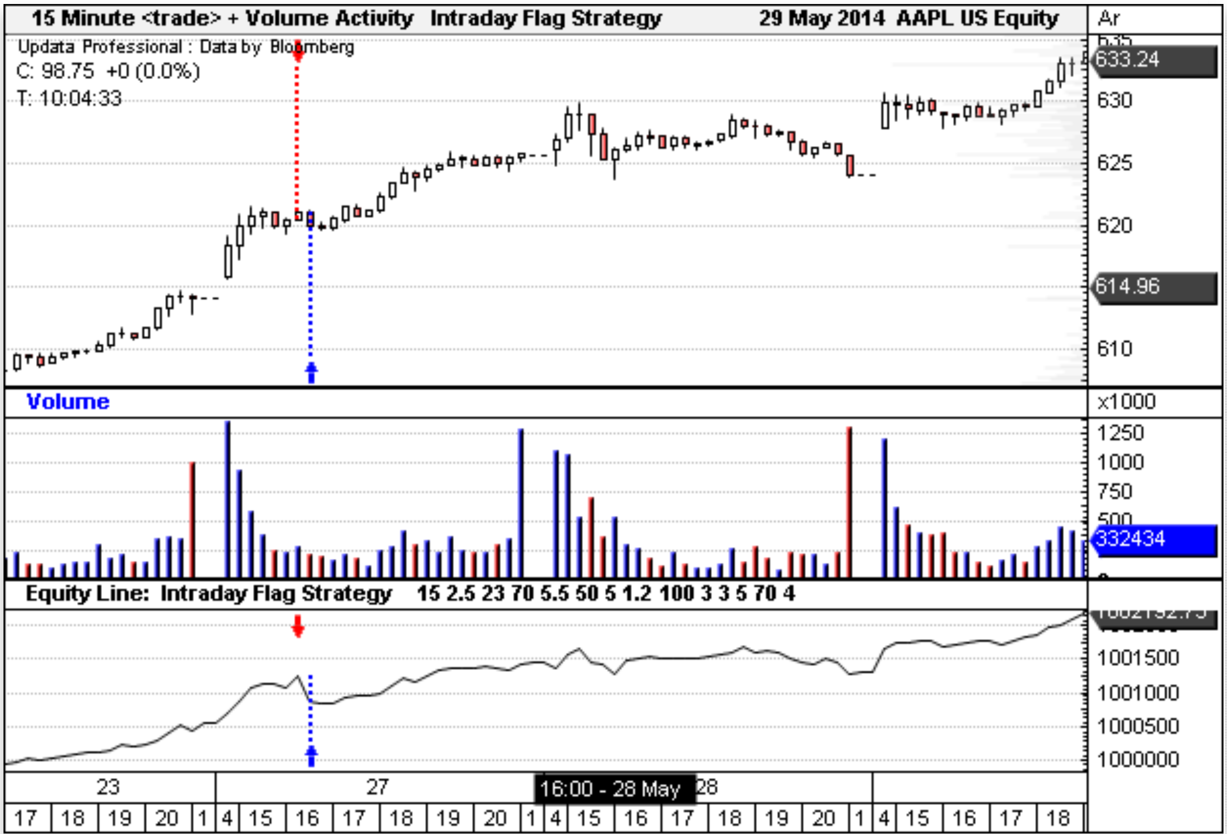


FIGURE 9: UPDATA. Here is an example of the intraday flag strategy applied to a chart of Apple Inc. (AAPL) in 15-minute resolution.

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

BACK TO LIST

AIQ: DECEMBER 2014

The AIQ code based on Markos Katsanos' article in this issue, "Detecting Flags In Intraday Charts," is provided at www.TradersEdgeSystems.com/traderstips.htm, and is also shown here:

```
!DETECTING FLAGS IN INTRADAY CHARTS

!Author: Markos Katsanos, TASC December 2014

!Coded by: Richard Denning 10/18/14


!USER DEFINED FUNCTIONS:

C is [close].

Name is description().


!COMPLETED FLAG PATTERN:

FLAG is [Flag].

FLAG_breakoutup if FLAG > 0.

FLAG_breakoutdn if FLAG < 0.


!EMERGING FLAG PATTERN:

e_FLAG is [eFLAG].

e_FLAGup if e_FLAG > 0.

e_FLAGdn if e_FLAG < 0.


!REPORTS TO LIST ALL FLAG PATTERS:

ShowAllCompleted if C>0 and FLAG <> 0.

ShowAllEmerging if C>0 and e_FLAG <>0.
```

The AIQ program has a chart-pattern recognition module that operates only in daily mode. I am providing code to find both completed flag patterns and also emerging flag patterns.

In Figure 10, I show a chart of G-III Apparel Group Ltd., which shows a flag pattern completed on June 25, 2014 (green up arrow), when the price broke above the down-sloping flag top. Although the volume was above average on the breakout, the followthrough was lacking.

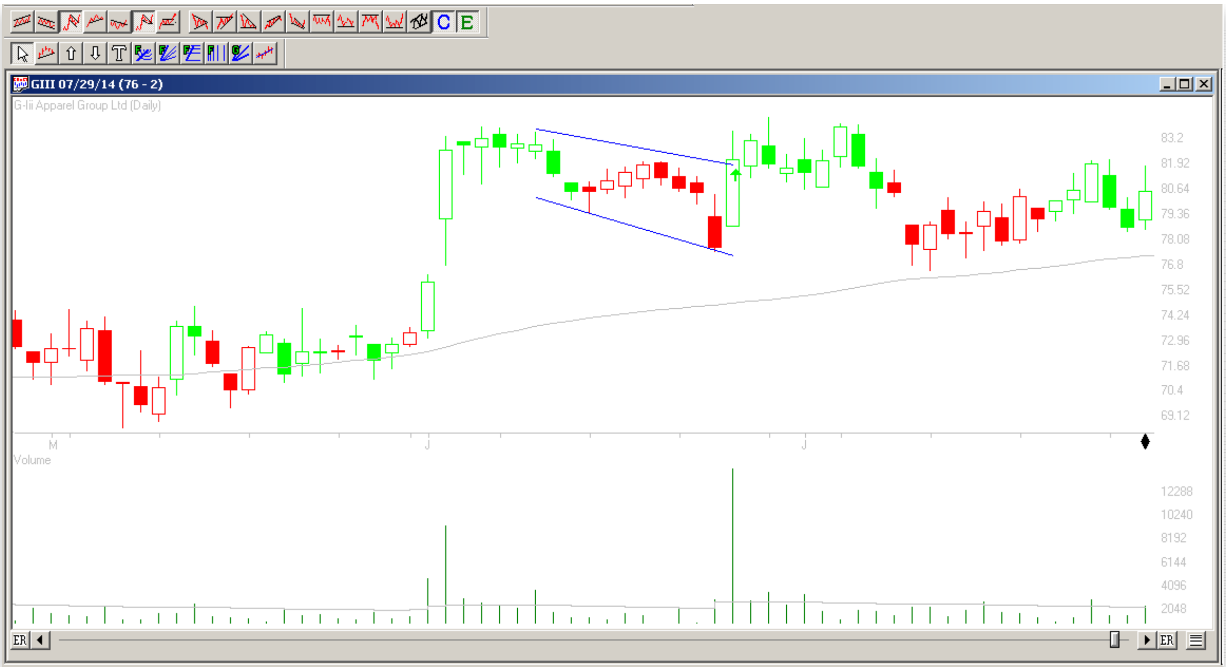


FIGURE 10: AIQ. This sample chart shows G-III Apparel Group Ltd. (GIII) with a completed flag pattern (indicated by the green up arrow).

Note that I did not code exits for the pattern, as the built-in exits can be used to experiment with the flag pattern entry. Note also that the AIQ version of flags does not match exactly the intraday flags that are defined by Katsanos in his article.

—Richard Denning
info@TradersEdgeSystems.com
for AIQ Systems

BACK TO LIST

TRADERSSTUDIO: DECEMBER 2014

The TradersStudio code based on Markos Katsanos' article in this issue, "Detecting Flags In Intraday Charts," 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 POLE: Returns a "1" if, based on the inputs, a sharp runup has occurred, otherwise returns a zero

- Function FLAG: Returns a “1” if, based on the inputs, a flag has formed, otherwise returns a zero
- Function FLAG_BREAKUP: Returns a “1” if, based on the inputs, a breakout from a flag has occurred on the current bar, otherwise returns a zero
- System HIGH_FLAG_sys: Long-only buy next bar at market after flag breakout, then exit using one of three exits (time, profit target, or stop-loss)

The code is also shown here:

```
'DETECTING FLAGS IN INTRADAY CHARTS
'Author: Markos Katsanos, TASC December 2014
'Coded by: Richard Denning 10/18/14
'www.adersEdgeSystems.com

Function POLE(atrPoleHeight,flagPoleLen,atrLen,ByRef poleHeight)
'atrPoleHeight=5.5,flagPoleLen,atrLen
poleHeight = Highest(H,flagPoleLen)-Lowest(L,flagPoleLen)
If poleHeight/avgtruerange(atrLen)>atrPoleHeight Then
    POLE = 1
Else POLE = 0
End If
End Function
'--------------------------------------------------------------------------------------
----------
Function FLAG(maxFlagLen,minFlagLen,maxFlagDepth,atrLen)
Dim poleHigh
Dim barsToHigh
poleHigh = Highest(H,maxFlagLen)
barsToHigh = MRO(H=poleHigh,maxFlagLen,1)
If countof(H > poleHigh,barsToHigh,0)=0 And barsToHigh >= minFlagLen Then
    If (Highest(H,maxFlagLen) - Lowest(L,maxFlagLen))/avgtruerange(atrLen) <=
maxFlagDepth Then
        FLAG = 1
    End If
Else
    FLAG = 0
End If
End Function
'--------------------------------------------------------------------------------------
-----------
Function FLAG_BREAKUP(maxFlagLen,minFlagLen,maxFlagDepth,flagPoleLen,atrLen)
'maxFlagLen=15,minFlagLen=5,maxFlagDepth=2.5,flagPoleLen=23,atrLen=50
If Highest(H,maxFlagLen+flagPoleLen,1)=Highest(H,flagPoleLen,1) Then
    If H > Highest(H,maxFlagLen,1) And
FLAG(maxFlagLen,minFlagLen,maxFlagDepth,atrLen) Then
        FLAG_BREAKUP = 1
    End If
Else
    FLAG_BREAKUP = 0
End If
End Function
'--------------------------------------------------------------------------------------
------------
'COUNTOF Function
'returns how many times a rule is true in the lookback length
'coded by Richard Denning 01/04/08
```

```
Function COUNTOF(rule As BarArray, countLen As Integer, offset As Integer)
Dim count As Integer
Dim counter As Integer
    For counter = 0 + offset To countLen + offset - 1
        If rule[counter] Then
            count = count + 1
        End If
    Next
COUNTOF = count
End Function
'-------------------------------------------------------------------------------------
-----------------------------------
Sub
HIGH_FLAG_sys(atrPoleHeight,flagPoleLen,atrLen,maxFlagLen,minFlagLen,maxFlagDepth,tren
dLen,maxBars,profitTargetMult)
'atrPoleHeight=5.5,flagPoleLen=23,atrLen=50,maxFlagLen=15,minFlagLen=5,maxFlagDepth=2.
5,maxBars=20,profitTargetMult=1.2
Dim isPOLE As BarArray
Dim isFlagBU As BarArray
Dim isUpTrend As BarArray
Dim higherVola As BarArray
Dim poleHeight As BarArray
isPOLE = POLE(atrPoleHeight,flagPoleLen,atrLen,poleHeight)
isFlagBU = FLAG_BREAKUP(maxFlagLen,minFlagLen,maxFlagDepth,flagPoleLen,atrLen)
isUpTrend = IIF(C>Average(C,trendLen),1,0)
higherVola = avgtruerange(atrLen) < Avgtruerange(maxFlagLen)
If isPOLE And isFlagBU And isUpTrend Then Buy("LE",1,0,Market,Day)
If BarsSinceEntry > maxBars Then ExitLong("LX_time","",1,0,Market,Day)
If C >= EntryPrice + poleHeight[BarsSinceEntry] Then
ExitLong("LX_pt","",1,0,Market,Day)
If C < Lowest(L,flagPoleLen,BarsSinceEntry) Then ExitLong("LX_sl","",1,0,Market,Day)
End Sub
'-------------------------------------------------------------------------------------
-------------------------------------
```

Note that the parameters were taken from the author’s intraday testing and may not be the desired ones for daily bar trading. Note also that this system cannot be used to trade intraday, as TradersStudio does not as yet have a real-time module, although historical intraday testing can be done on saved intraday data.

In Figure 11, I show a chart of Autodesk (ADSK) with a flag breakout trade that exited at the profit target.

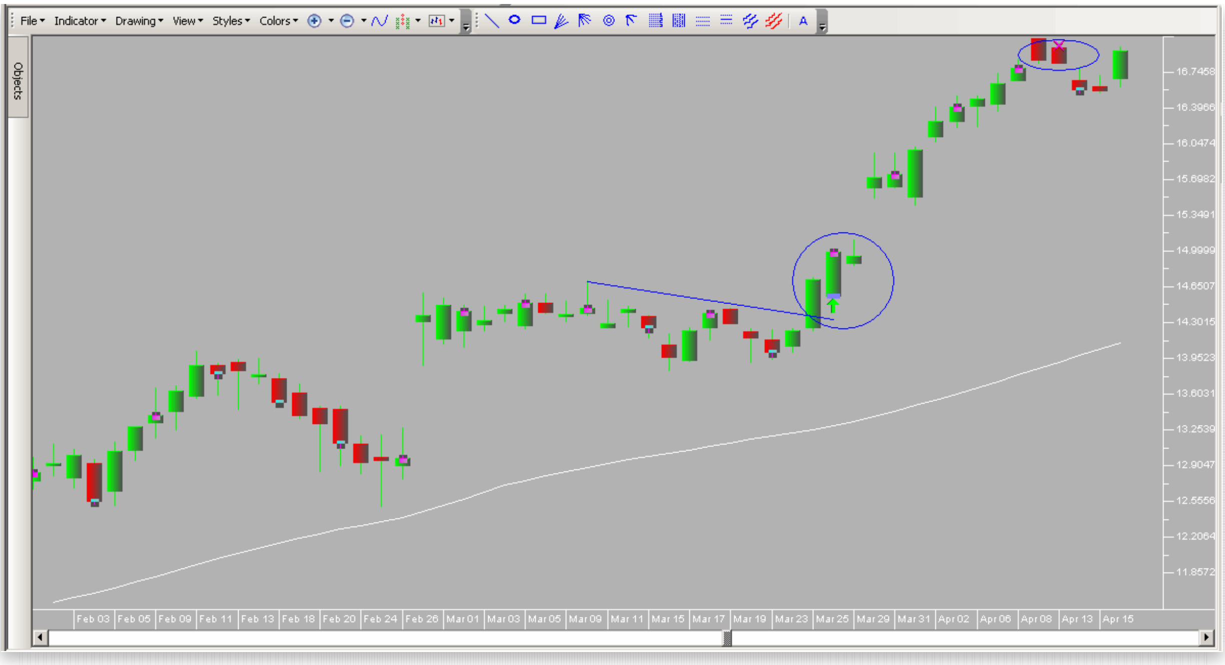


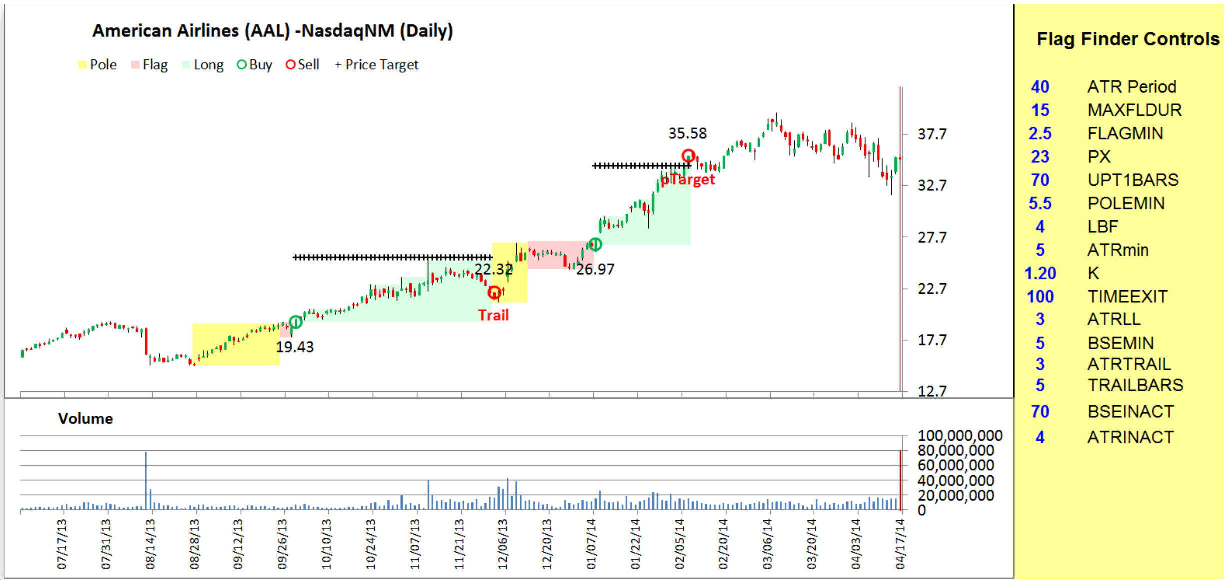
FIGURE 11: TRADERSSTUDIO. Here is an example flag breakout trade that reached the profit target on Autodesk (ADSK).

—Richard Denning
info@TradersEdgeSystems.com
for TradersStudio

BACK TO LIST

MICROSOFT EXCEL: DECEMBER 2014

The techniques presented in Markos Katsanos' article in this issue, "Detecting Flags In Intraday Charts, can be applied at tick, end-of-day, weekly, and longer bar-size intervals. For ease of data access, I have used American Airlines (AAL) end-of-day data for this example (Figure 12).



Flag Finder Controls

40	ATR Period
15	MAXFLDUR
2.5	FLAGMIN
23	PX
70	UPT1BARS
5.5	POLEMIN
4	LBF
5	ATRmin
1.20	K
100	TIMEEXIT
3	ATRLL
5	BSEMIN
3	ATRTRAIL
5	TRAILBARS
70	BSEINACT
4	ATRINACT

FIGURE 12: EXCEL, TRADE COMBINATIONS. Price chart with two pole, flag, and trade combinations. A short LBF falue found these.

The 16 user control values used in the article provide a lot of opportunities for exploration. For example, here, an LBF of 4 finds two adjacent profitable flags. LBF=50 as shown in the article only finds one of these.

K, the pole multiplier used to set the price target, also can have significant impact on trades and the location of additional examples.

As built, this spreadsheet will not highlight pole and flag combinations that do not lead to a breakout and trade initiation.

Also, as built, the *high* of a bar may exceed the target price, but the trade may not exit. The logic provided in the article requires that the close of the bar must exceed the target price to initiate a price target close.

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 ("DetectingFlags.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 December 2014 issue of