Templates Manual

Ensign 10

Last update 6 August 2013

Table of Contents

Ensign Templates

Overview

Click the Templates button to open the Templates panel. Templates are used to apply multiple Studies to a chart at one time. For example, a Template could be created that adds 3 studies to the chart at the same time. This is much faster than applying each study one at a time.

Creating and Saving a Template

To create a new template, first open a chart and apply all the studies that you want to include in the template. Set the parameters and colors of each study to your preferences. Next, click the templates button to display the Templates panel.
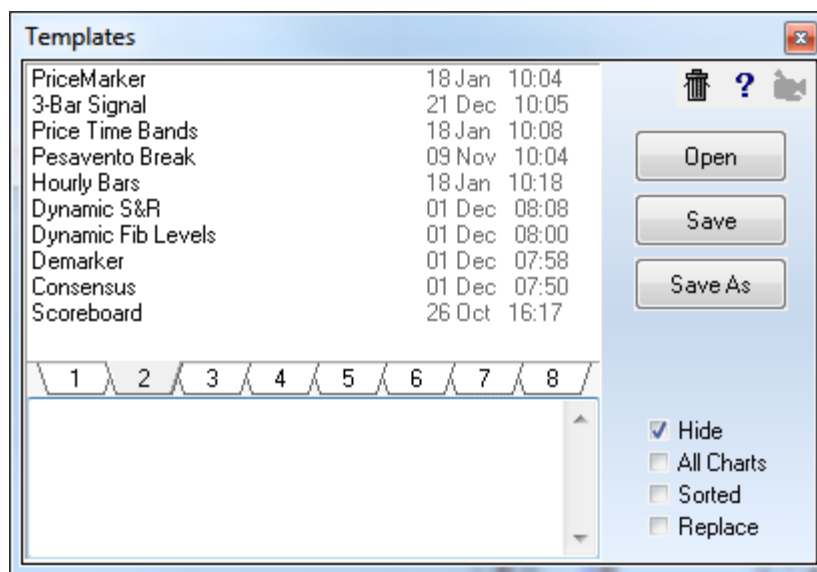
Click the Save As button, and then enter a Template name in the entry box. The template name will be added to the list box. There is no limit to the number of templates that can be saved. If you make changes to a template, click the Save button to resave the template after making changes to a chart. To save a template under a different name click on the Save As button and enter a new template name.

The Template will remember every detail regarding each study. Draw Lines that are placed in study sub-windows are also remembered. Note: A Template will not remember Draw Tools that are placed on a chart (except 'Daily Price Lines') since each chart is different.

Super Template

A super template will contain additional information for setting the chart's 15 color properties, candlestick properties, grid style, and margins.   To create a super template, use a '+' character in the name of the template when saving the template.

Using Templates

Templates are applied to a chart by selecting the Template name in the list box and then clicking Open. Double-clicking on the Template name will also open the Template. The All Charts check box specifies whether the template is applied to all open charts, or just the active chart window. If the All Charts check box is checked, then the selected template is applied to all open charts. If the All Charts check box is unchecked, then the template is only applied to the active chart.

Hide - Check the 'Hide' box to specify auto-closing of the Templates panel after a template is opened. Uncheck the box to have the panel remain open so that other templates can be quickly opened.

Sorted - Place a check mark in the Sorted check box to sort the templates in alphabetical order. Uncheck the box to arrange the names in a custom ordering.

Replace - When the Replace box is checked, the template being applied to the chart will replace the studies that are currently on the chart. Uncheck the box to have the template added to the studies on the chart.

Deleting and Repositioning Templates

To delete a template from the templates list, click the templates button and then select the template name in the list box. Click the Delete button (Trash Can image) to remove the template name from the list box.

Drag a template name vertically up or down on the list to rearrange the order of the list.

At the bottom of the template list are 8 tabs for selecting a template folder. The tabs permit templates to be read and saved in different folders. The template folders are named \Template# where # is a digit 1 through 8. You might use one folder for templates under development, and another folder for permanent templates.
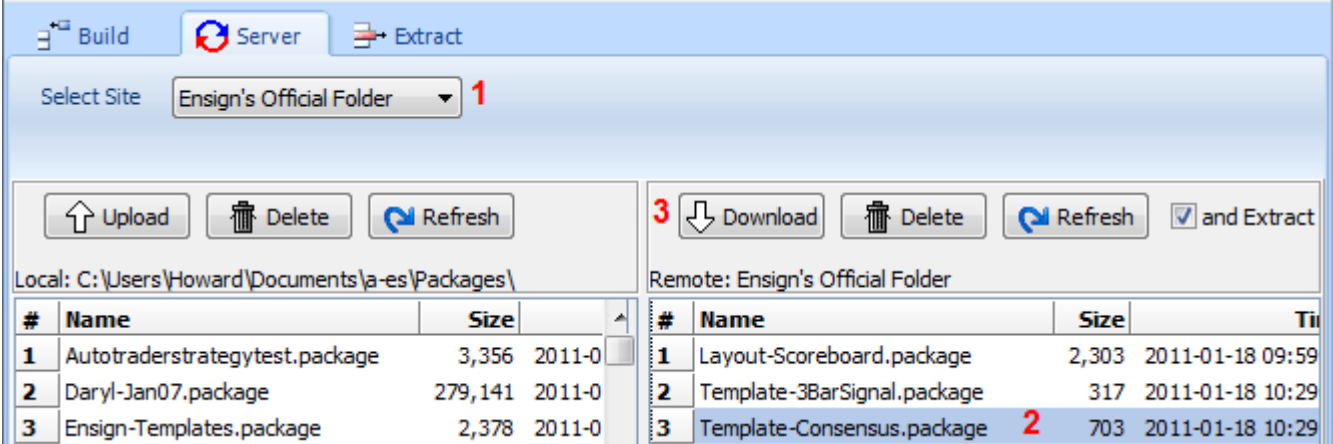
Template files can be moved from one folder to another by dragging a template name from the list to the new tab at the bottom of the list.

Sharing Templates with other Traders

If desired, template files can be emailed to other Ensign users. This allows traders to share their templates with associates and friends. Templates are saved in text files in the \Ensign10 \Templates\Template# folder, where # is a digit 1 through 8. The file name extension is .tmp (example: MyTemplate.tmp). When a template file is saved in the \Template1 folder, the new template will appear in the tab 1 templates list the next time the template panel is opened. The easiest way to distribute a template is to include it in a Package, and upload it to the Ensign web server.

Downloading Templates from Ensign's Web Site

Click the Setup | Package button. Select the Server tab and the Ensign's Official Folder.   The templates created by Ensign Software will be in this folder and begin with the word Template.

Templates distributed by other traders will probably be located in General Public Folder.

Select one or more packages to download from the server and click the Download button. Check the 'and Extract' box before downloading so the extraction of the templates will be performed automatically after the packages are downloaded. The extraction puts the template files in the template folder they were packaged from. Look for Ensign's templates in the 2nd template folder.

Descriptions

60 Minute Support and Resistance

Three chart prices are used to calculate Support and Resistance lines. Typically the High, Low, and Close prices of a daily range are used. However, this DYO example will use 3 prices from the prior 60 minute period. The 3 prices will be the high, low and close of the prior hour.

Formula

Pivot   = (High Price + Low Price + Close Price) / 3
Range = High Price – Low Price

4R [4th Resistance] = 2R + Range
3R [3rd Resistance] = 1R + Range
2R [2nd Resistance] = Pivot + Range
1R [1st Resistance] = Pivot + (Pivot - Low)
H   [1st Hour High]
P   [Pivot Price]
L   [1st Hour Low]
1S [1st Support] = Pivot - (High-Pivot)
2S [2nd Support] = Pivot - Range
3S [3rd Support] = 1S – Range
4S [4th Support] = 2S – Range
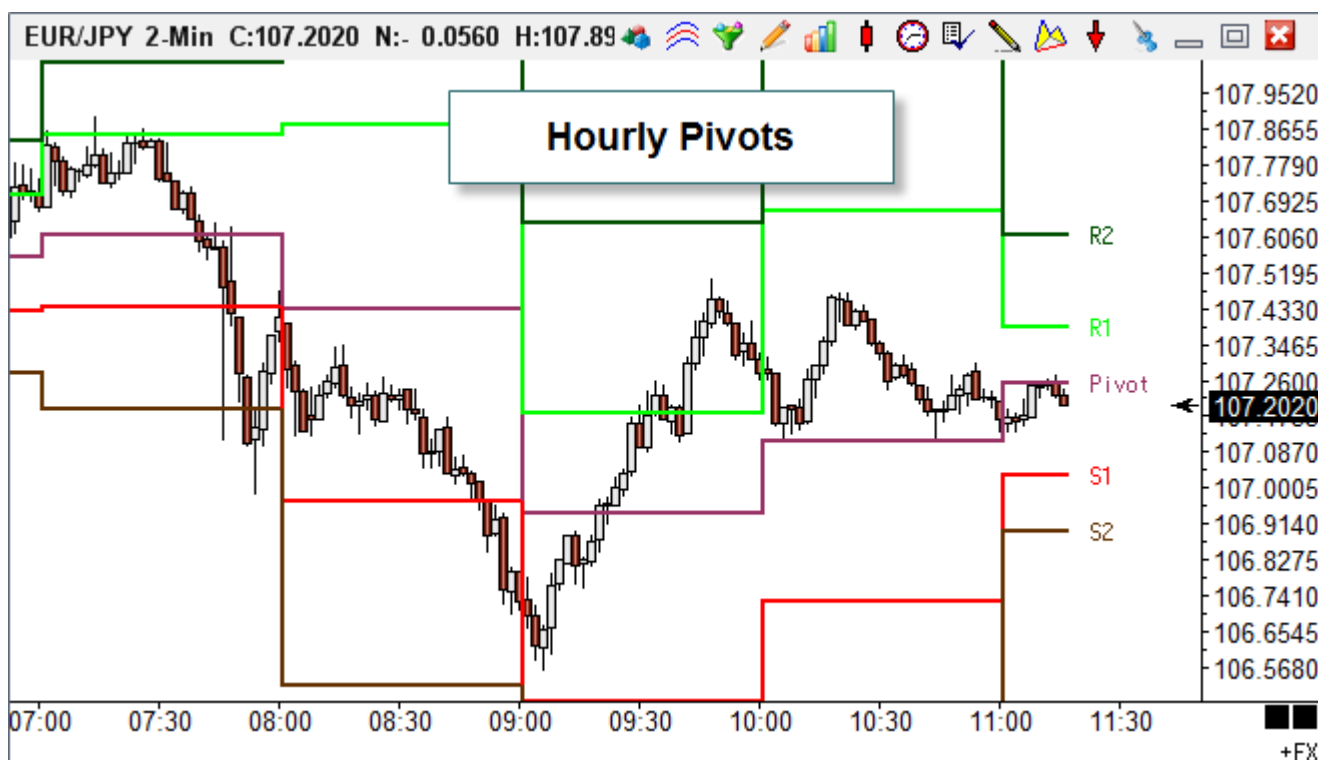
Study Name: 60 min S & R Lines
Study Location: Chart
Marker Location: Study Value
Label Location: Margin 2
Grid Tab: Default

Study Scale: Chart Scale
Variables File: -60minS&R
Message Location: None

Message Text: [blank]
Font: ■ True: ■ False: ■ Panel: ■

Sound
● Silent   ○ Beep   ○ Voice   ○ WAV   ☐ Sound once per bar
WAV file: [blank]   [Browse]

Trading System
Price: Close
Quantity: 1

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show
Function	56 Pivot	= Sum(#2, [#])	3	[Close]	0	✓
	/	(3			0	) Pivot

A	Expression	[Period] := 60	☐				
B	Chart Value	[High] := Prior [Period] Minute High	☐				
C	Chart Value	[Low] := Prior [Period] Minute Low	☐				
D	Chart Value	[Close] := Prior [Period] Minute Close	☐				
E	Function	[Pivot] := Sum([Close], 3) / 3	✓	⌐	■	Study Value	Pivot
F	Expression	[S1] := (2 * [Pivot]) - [High]	✓	⌐	■	Study Value	S1
G	Expression	[S2] := [Pivot] - ([High] - [Low])	✓	⌐	■	Study Value	S2
H	Expression	[R1] := (2 * [Pivot]) - [Low]	✓	⌐	■	Study Value	R1
I	Expression	[R2] := [Pivot] + ([High] - [Low])	✓	⌐	■	Study Value	R2

A – 60 is the number of minutes in the period.   This can be changed for other periods.

B – Save the high of the prior time period.   The duration of the period is from row A.

C – Save the low of the prior time period in a variable named Low.

D – Save the close of the prior time period in a variable named Close.

E – Calculate the Pivot by averaging [Close], [High], and [Low]. The 3 variables used are adjacent to each other with [Close] being first on the variable list. This is an important relationship because the function averages adjacent variables with the variable which is 1st in the list being named in the function parameter. The quantity of 3 comes from the Op. [#] field.

F – Calculate 1st Support.
G – Calculate 2nd Support.
H – Calculate 1st Resistance.
I – Calculate 2nd Resistance.

The line labels use Label Location Column 3 so they show in the right side margin.

3-Bar Signal

This example implements the following Sell Signal and Buy Signal.

Rules for the Sell Signal:

1. Bar 2 has a higher high and higher low than both bars 1 and 3.
2. Bar 1 is a green bar (close higher than open).
3. Bar 3 is a red bar (close below open or close = open).
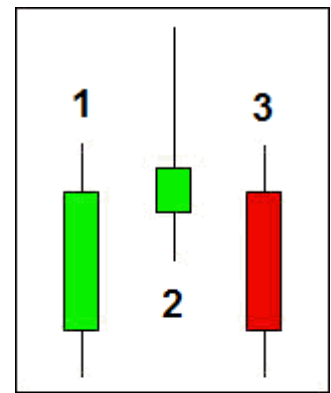
Rules for the Buy Signal:

1. Bar 2 has a lower high and lower low than both bars 1 and 3.
2. Bar 1 is a red bar (close lower than open).
3. Bar 3 is a green bar (close above open or close = open).

		Study Name	Study Location	Marker Location	Label Location	Grid Tab	
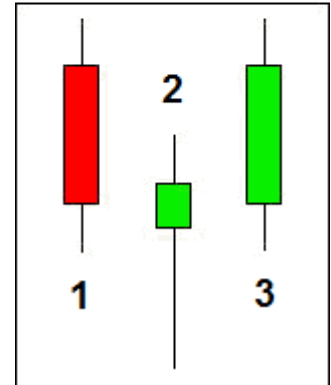
Study Name: 3-Bar Signal
Study Location: Chart
Marker Location: Above High 1
Label Location: None
Grid Tab: Default

☐ Use as Default
☐ Draw Behind Bars
☐ Privatize
☐ Close Only
☐ Show Values
☐ Email
☐ Auto Remove
☐ Auto Completion

Study Scale: Chart Scale
Variables File: -3BarSignal
Message Location: None

Message Text: [ ] Font: ■ True: ■(green) False: ■(red) Panel: ■(yellow)

Sound
◉ Silent ○ Beep ○ Voice ○ WAV ☐ Sound once per bar
WAV file: [ ] [Browse]

Trading System
Price: [ ] Quantity: 1 Commission: 0

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
Expression ▾	20 Rule 1a ▾	= High ▾	> ▾	Prior High ▾	-1	✓	• ▾	▨
	AND ▾	(Low ▾	> ▾	Prior Low ▾	-1			)

|	Category	Expression	Show	Marker	Color	Location	
A	Expression	[Rule 1a] := (High[-1] > Prior High[-1]) AND (Low[-1] > Prior Low[-1])	✓	•		Above High	
B	Expression	[Rule 1b] := (Prior High > High) AND (Prior Low > Low)	✓	•		Above High	
C	Expression	[Rule 2] := Close[-2] > Open[-2]	✓	•		Above High	
D	Expression	[Rule 3] := Close <= Open	☐				
E	Flag	[0] := AND([Rule 1a], 4)	✓	⬤	▮	Above High	
F			☐				
G	Expression	[Rule 1a] := (High[-1] < Prior High[-1]) AND (Low[-1] < Prior Low[-1])	✓	•		Below Low	
H	Expression	[Rule 1b] := (Prior High < High) AND (Prior Low < Low)	✓	•		Below Low	
I	Expression	[Rule 2] := Close[-2] < Open[-2]	✓	•		Below Low	
J	Expression	[Rule 3] := Close >= Open	☐				
K	Flag	[0] := AND([Rule 1a], 4)	✓	⬤		Below Low	

A – Implements Rule 1: Bar 2 has a higher high and a higher low than Bar 1.

B – Implements Rule 1: Bar 2 has a higher high and a higher low than Bar 3.

C – Rule 2: Bar 1 Close is above its open, ie. a Green bar.

D – Rule 3: Bar 3 Close is <= its Open, ie. a Red bar.

E – Show a marker when all 4 conditions are simultaneously True. This is the Sell Signal.

Note: Rule1a, Rule1b, Rule2 and Rule3 are Variables in sequence beginning with Rule1a.

Rows G through K implement the same principles for the Buy Signal.

Acceleration Bands

Acceleration Bands were created by Price Headley (http://www.bigtrends.com/).

Acceleration Bands are an envelope of the chart's normal volatility over a period of time (normally 20 bars). Weekly and Monthly charts, according to Headley's use, use the bands as an indication of breakouts and lower time frames as Support and Resistance.

Two consecutive closes above the top band, or two consecutive closes below the bottom band signals a possible turning point. The chart shows a red ball where two consecutive closes occurred above the top band, and a green ball where two consecutive closes occurred below the bottom band.
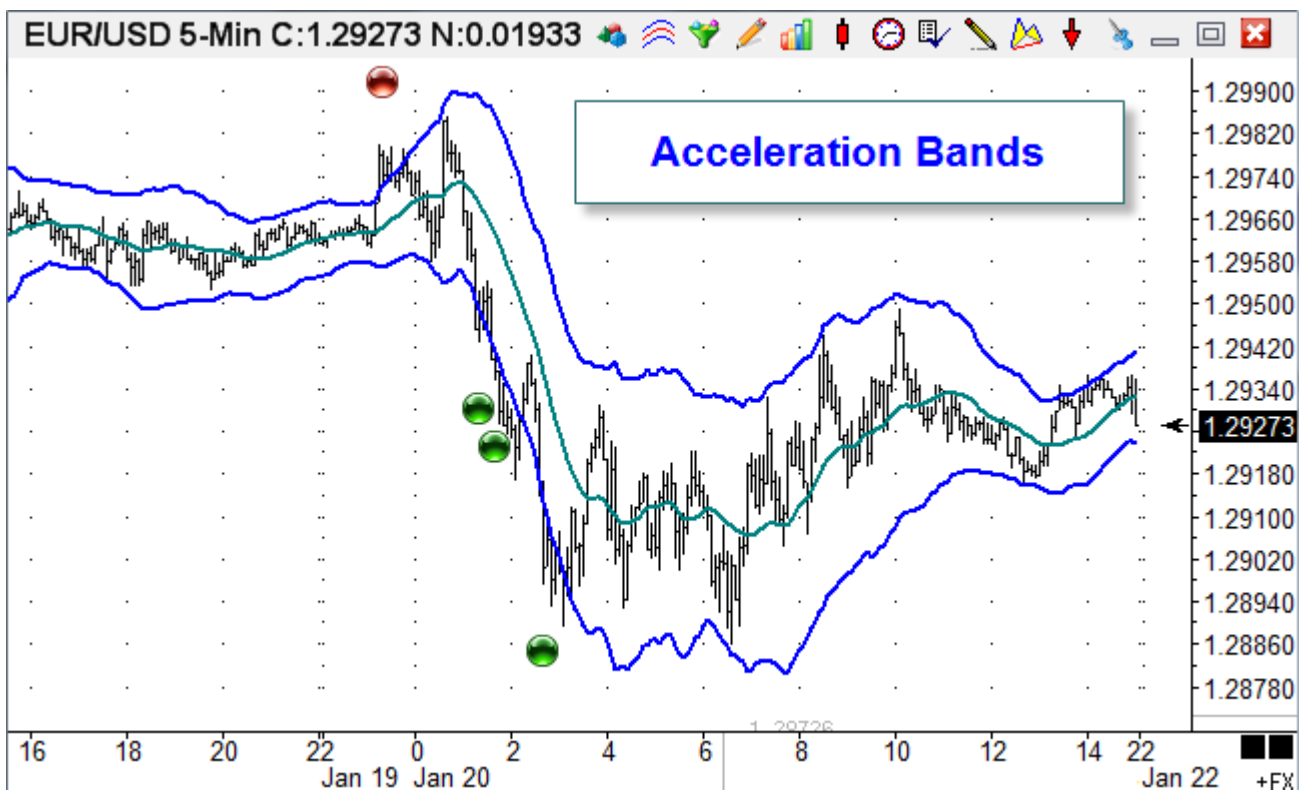
Formula

Factor = 0.001

High Band = (High * (1 + 2 ((((High - Low) / ((High + Low) / 2)) * 1000) * Factor)))

Low Band = (Low * (1 - 2 ((((High / Low) / ((High + Low) / 2)) * 1000) * Factor)))

Midpoint = (High Band + Low band) / 2

A	Expression	[Period] := 20		
B	Expression	[x1000] := (Range / (High+Low)/2) * 1000		
C	Expression	[xDec] := ([x1000] * 0.001) * 2		
D	Expression	[Upper Value] := (1 + [xDec]) * High		
E	Expression	[Lower Value] := (1 - [xDec]) * Low		
F	Function	[Upper Band] := Simple Average([D], [Period])	✓ ⌢ ■	Study Value
G	Function	[Lower Band] := Simple Average([E], [Period])	✓ ⌢ ■	Study Value
H	Expression	[MidPoint] := ([Upper Band] + [Lower Band]) / 2	✓ ⌢ ■	Study Value
I	Expression	[0] := (Close[-1] X> [F][-1]) AND (Close > [F])	✓ ●	Above High
J	Expression	[0] := (Close[-1] X< [G][-1]) AND (Close < [G])	✓ ●	Below Low

A – Define the Period for the Simple Averages on row F and G.

B – Calculate (High - Low) / ((High + Low) / 2)) * 1000

C – Calculate the rest of the decimal adjustment for the multiplier.

D – Calculate the Upper Value which is the High times a multiplier.

E – Calculate the Lower Value which is the Low times a multiplier.

F – Plot a 20 Period Simple Moving Average of the Upper Values on row D.

G – Plot a 20 Period Simple Moving Average of the Lower Values on row E.

H – Plot the Midpoint of the Upper and Lower Bands

I – Put a marker on the trigger for the Upper Band. Prior close crossed above the band and current close is still above the band.

J – Put a marker on the trigger for the Lower Band. Prior close crossed below the band and current close is still below the band.

A variation of this template is available in the Template-AccelerationBandsHHLL package. The difference is that the Highest High and Lowest Low of the period are used in the formulas for the High and Low entries.

Arms' Ease of Movement Value

The Arms' Ease of Movement Value (EMV) is an indicator based on Momentum which qualifies volume and price changes and tries to determine the ease the price is to move up or down.

An optimized moving average crossing above zero is a buy trigger (and is highlighted in green) and crossing below zero is a sell trigger (and highlighted in red).

The DYO uses a simple moving average with a period of 9.

If the EMV exceeds 60% then there is 'Extreme Optimism' in the ease. If EMV drops below 40% then there is 'Extreme Pessimish' in the ease.   Black dots in the sub-window signify these areas.
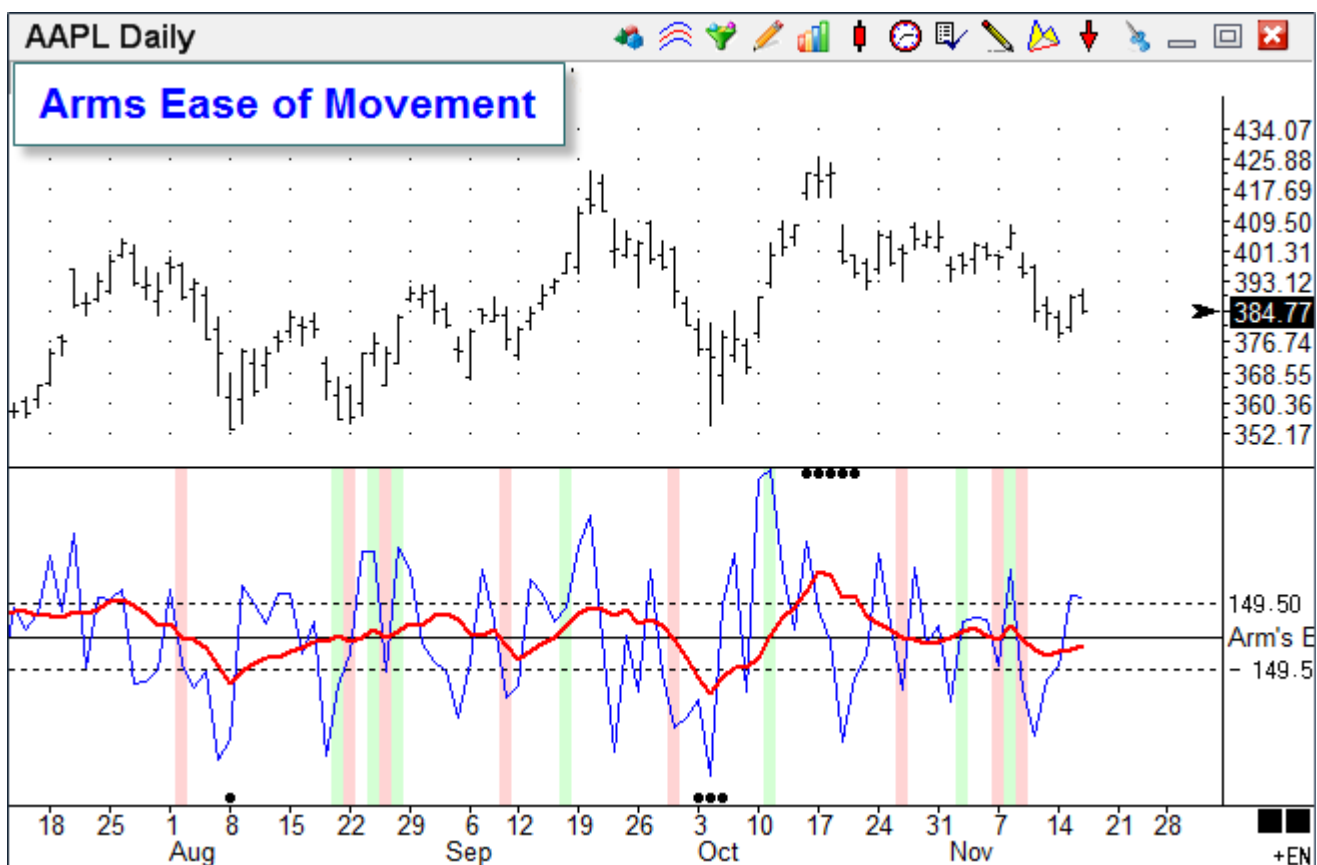
Formula

EMV = ((High+Low)/2 – (Prior High+Prior Low)/2) / Volume / (High – Low)

Study Name: Arms Ease of Movement
Study Location: Window 1
Marker Location: Study Value
Label Location: None
Grid Tab: Default

Study Scale: Data 0 Centerline
Variables File: -ArmsEase
Message Location: None

Message Text:
Font / True / False / Panel

Sound: ◉ Silent ○ Beep ○ Voice ○ WAV ☐ Sound once per bar
WAV file: [Browse]

Trading System
Price: Close
Quantity: 1

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Expression	102 EMV	= (High+Low)/2	-	Prior (H+L)/2	0
	/	(Volume	/	Range	0)

A	Expression	[Period] := 9	☐			
B	Expression	[Multiplier] := 1000000000	☐			
C			☐			
D	Expression	[EMV] := ((High+Low)/2 - Prior (H+L)/2) / (Volume / Range)	☐			
E	Expression	[EVMMult] := [EMV] * [Multiplier]	☑	⌒	(blue)	Study Value
F	Function	[Average] := Simple Average([E], [Period])	☑	⌒	(red)	Study Value
G			☐			
H	Flag	[Buy] := [F] X> 0	☑	ZON	(green)	Study Value
I	Flag	[Sell] := [F] X< 0	☑	ZON	(pink)	Study Value
J	Function	[0] := [Average] as % of Sub-window Scale > 60	☑	•	(black)	Top Row 1
K	Function	[0] := [Average] as % of Sub-window Scale < 40	☑	•	(black)	Bottom Row

A – Define period for the moving average on Line F.

B – Define a multiplier to counter the magnitude of the volume division on Line D.

D – Implement the EMV formula.

E - Rescale the EMV result by the multiplier. Optionally plots the EMV as a blue line.

F – Create a Simple Moving Average of the EMV and plot as a red line.

H, I – Show the Buy and Sell triggers for when the Average crosses zero.

J, K – Show when the Average is above 60 percent or below 40 percent of the scale.

Average Day Range

This example uses a spreadsheet to show the daily average range for several currencies. Both the 5 day average range and the 365 day average range are shown.

Spreadsheet	ADR	
Pair	5 day ave	365 day ave
1		
2 AUD/CAD	0.0069	0.0094
3 AUD/CHF	0.0089	0.0131
4 AUD/JPY	0.7200	1.1449
5 AUD/NZD	0.0080	0.0107
6 AUD/USD	0.0105	0.0131
7 CAD/CHF	0.0088	0.0123
8 CAD/JPY	0.5808	1.0382
9 EUR/CAD	0.0115	0.0134
10 EUR/CHF	0.0042	0.0135
11 EUR/GBP	0.0069	0.0075
12 EUR/JPY	1.019	1.386
13 EUR/USD	0.01499	0.01566
14 GBP/AUD	0.0125	0.0171
15 GBP/CAD	0.0101	0.0151
16 GBP/CHF	0.0128	0.0178
17 GBP/USD	0.0124	0.0141
18 NZD/CAD	0.0068	0.0084
19 USD/CAD	0.0085	0.0099
20 USD/CHF	0.0111	0.0129
21 USD/JPY	0.3278	0.7957

The data is calculated on a set of daily charts organized in a stack. Each chart has a DYO that calculates the averages and posts the results on the spreadsheet. Here is an example:

A	Expression	[Row] := 5	☐				
B	Expression	[Decimals] := Scale Factor	☐				
C	Expression	[Week] := 5	☐				
D	Expression	[Year] := 365	☐				
E	Function	[0] := Simple Average(Range, [Week])	☐				
F	Function	[0] := Simple Average(Range, [Year])	☐				
G			☐				
H	Action	Find Spreadsheet('ADR')	☐			ADR	
I	Action	if True then Cell(1, [Row]) := '[$S]'	☐			[$S]	
J	Action	Cell(2, [Row]) := Format([E], [Decimals])	☐				
K	Action	Cell(3, [Row]) := Format([F], [Decimals])	☐				

A – Define a Row variable. Edit this value to change the row location on the spreadsheet.

B – Define a variable with the number of decimals. This value is used to format the values.

C, D – Define variables with the periods for the two averages.

E, F – Calculate simple averages using Range as the data point. The 2nd parameter is the period.

H – Locate the Spreadsheet with the 'ADR' selection.

I – Post the chart's symbol to the 1st column of the spreadsheet.

J – Post the 1st average from Row E and format the displayed result for the number of decimals to show. The average posts in the 2nd column.

K - Post the 2nd average from Row F, formatted for decimal display, and show in column 3.

Download the Template-AverageDayRange package using the Setup | Package feature. The package will install an example template to use, and the Average Range layout.   Opening the layout will create the following stack, and show the spreadsheet with the average ranges.
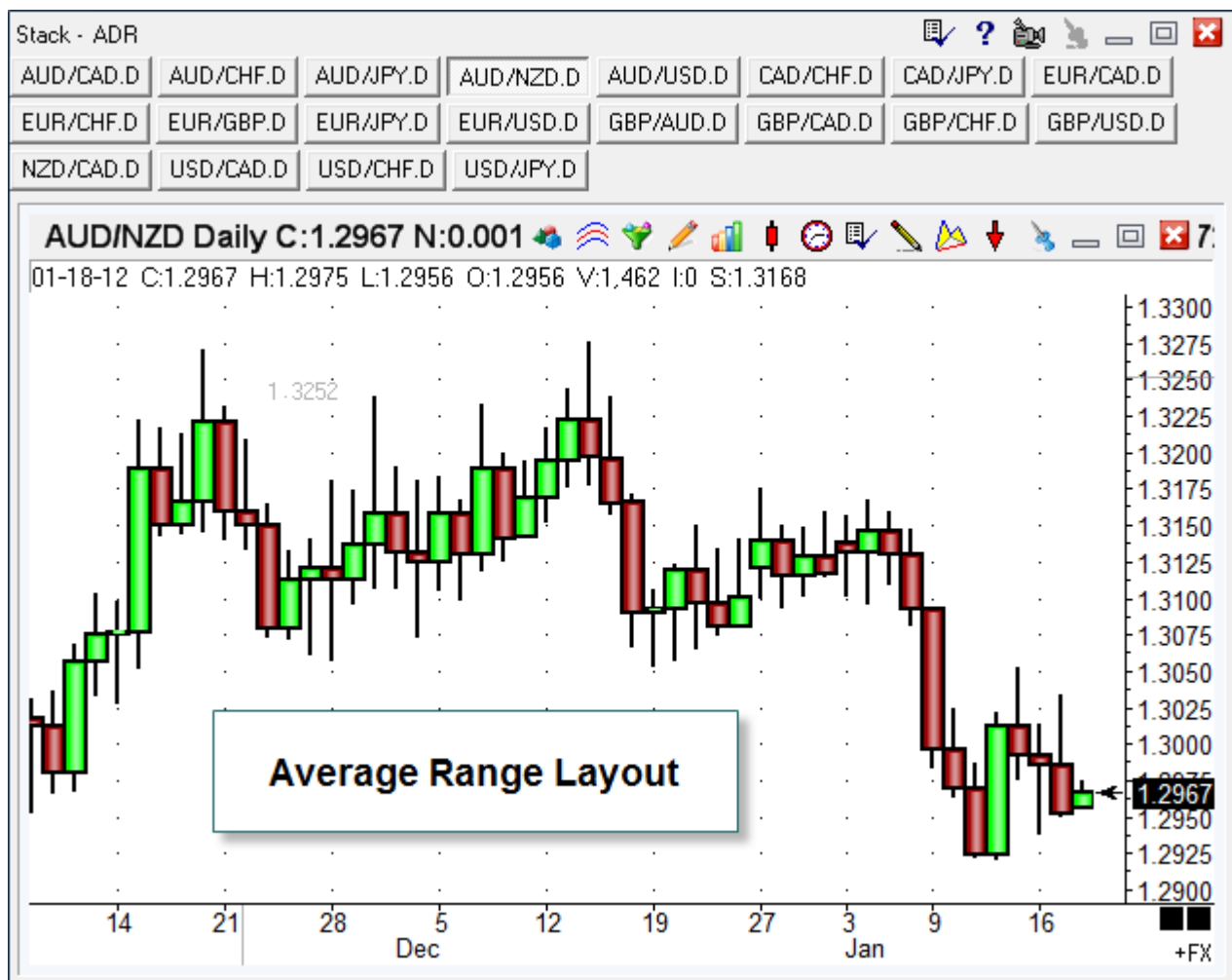
Camarilla Pivots

The Camarilla Pivots template implements the Camarilla formulas for showing support and resistance levels based on Yesterday's daily Close, High and Low prices.

Formula

R4 = C + RANGE * 1.1 /2
R3 = C + RANGE * 1.1 /4
R2 = C + RANGE * 1.1 /6
R1 = C + RANGE * 1.1 /12
PP = (HIGH + LOW + CLOSE) / 3
S1 = C - RANGE * 1.1 /12
S2 = C - RANGE * 1.1 /6
S3 = C - RANGE * 1.1 /4
S4 = C - RANGE * 1.1 /2
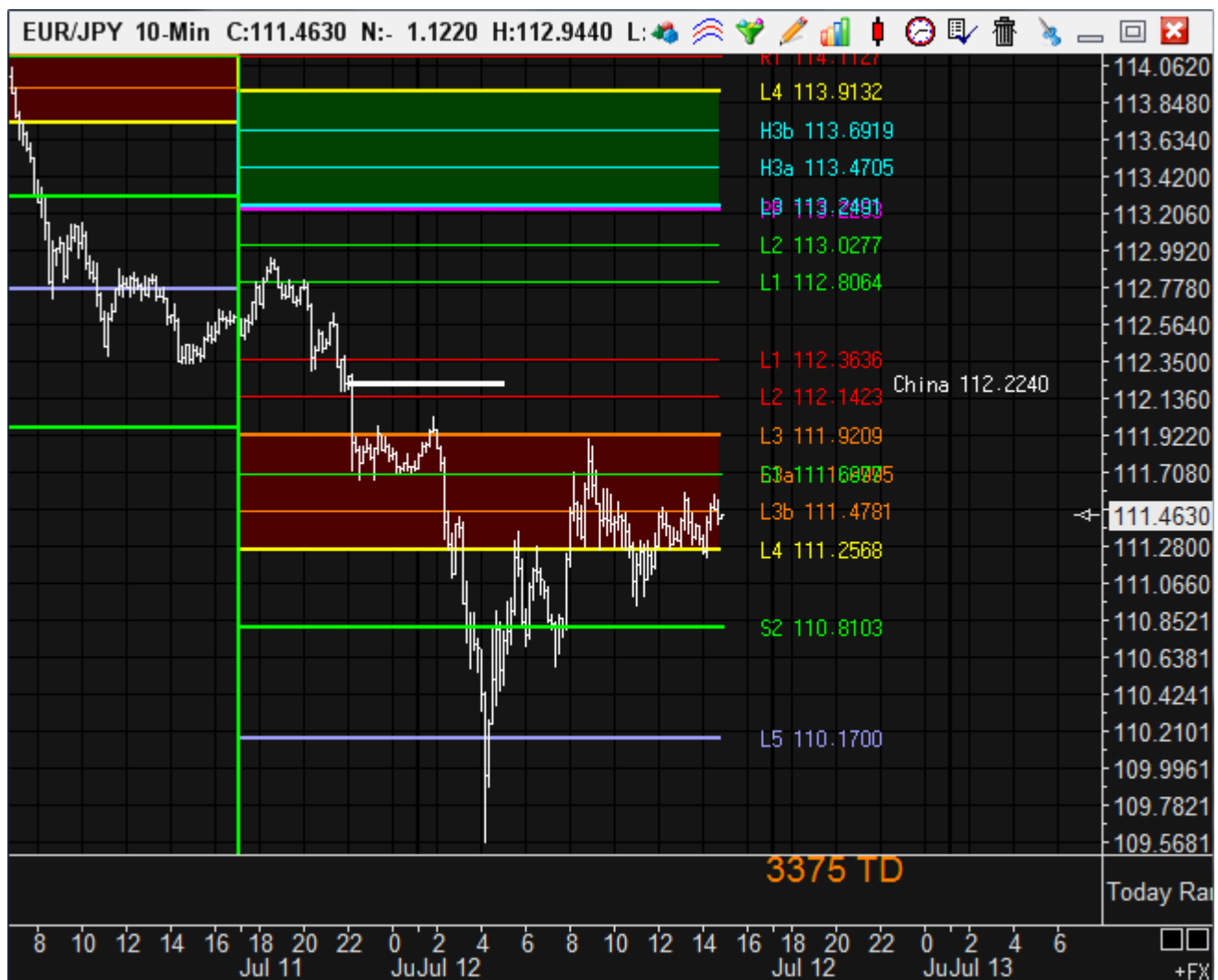
The template also marks when the Asian session is open and its opening price, since the Asian markets are a factor in Forex trading.

Conditional Donchian Plot

The Donchian Channel study plots with 2 channel lines forming a band. This DYO example will choose which channel line to plot based on the MACD study. When the MACD is above zero, the lower channel line is plotted. When the MACD is below zero, the upper channel line is plotted. The chart has a MACD, Donchian Channel, and DYO in this template.
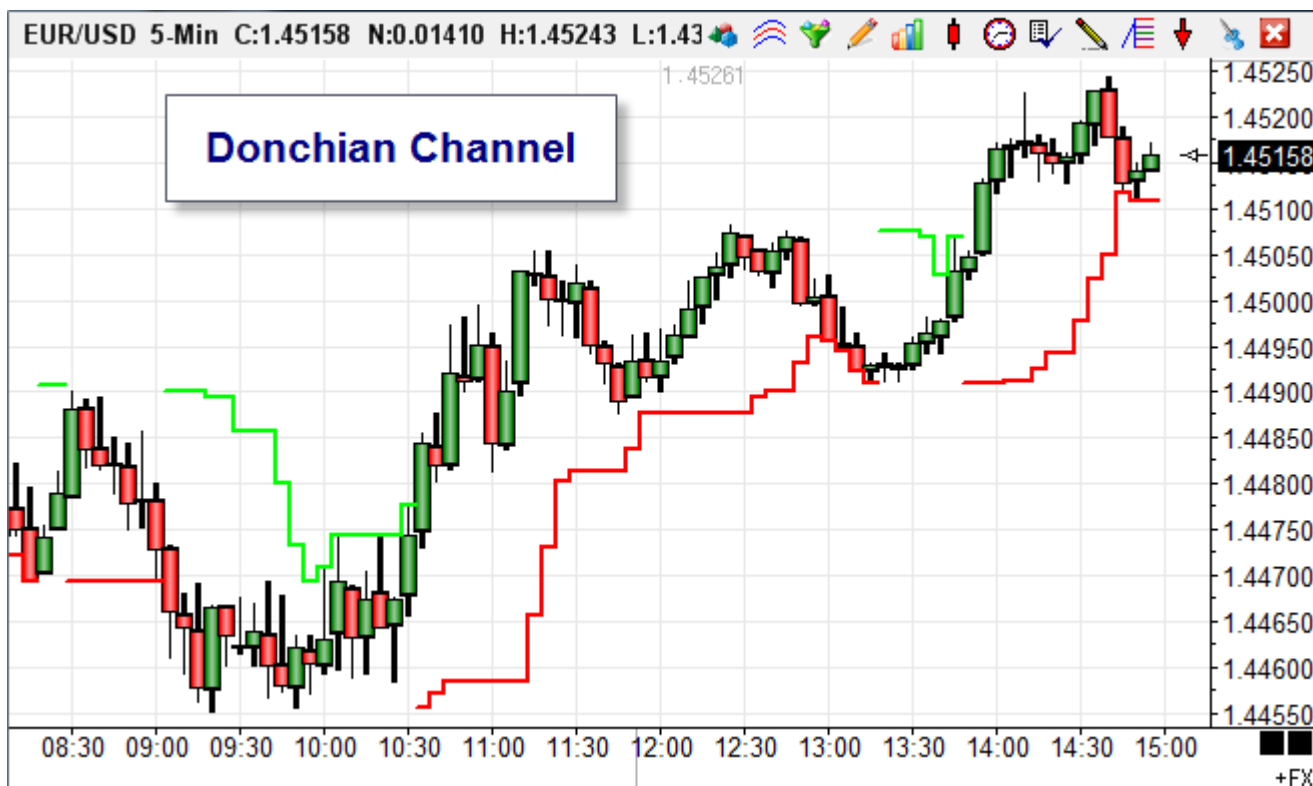
Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4		Offset
Action ▼	0 ▼	= V := ##; if #2 oper 0 then pl ▼	>= ▼	[20 MACD]	▼	0 ⬍
⬚ ⬚ ⬚		▼ ([22 LowerBand] ▼	▼		▼	0 ⬍)

A	Study	[20 MACD] := MACD.MACD value	☐		
B	Study	[21 UpperBand] := DON.1st Upper band	☐		
C	Study	[22 LowerBand] := DON.1st Lower band	☐		
D	Action	[0] := ([22 LowerBand]); if [20 MACD] >= 0 then plot V	☑ ⌐	🟥	Study Value
E	Action	[0] := ([21 UpperBand]); if [20 MACD] < 0 then plot V	☑ ⌐	🟩	Study Value

A, B, C – Read the study values into variables [20], [21], [22] for use by Rows D and E.

D – Return the Lower Band value, but only plot this value when MACD is above zero.

E – Return the Upper Band value, but only plot this value when MACD is below zero.

Note: Plotting of the Donchian channel lines is done by this DYO and not by the Donchian study. The Show boxes on the Donchian study property form are unchecked.

Cumulative Volume

This example keeps a running total of the volume at the Ask minus the volume at the Bid.

A - Zero out the variable that holds the sum at the start of the calculation run.

B & C - Plot the VolSum variable value using the 2 color fill marker.

DeMark Count

Tom DeMark describes this trading tool as looking for at least nine consecutive closes less than the close four trading bars earlier for a buy setup and at least nine consecutive closes greater than the close four trading bars earlier for a sell setup. This establishes a context that helps a trader determine whether they should be buying or selling the market. This tool is designed to anticipate a trend reversal.
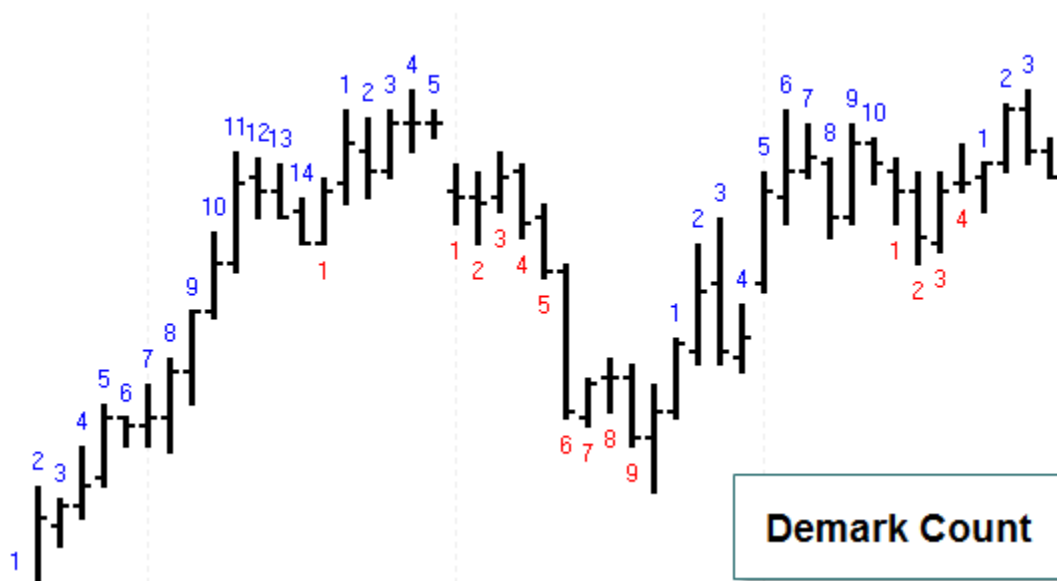
The indicator counts the bars in the sell setup and labels the count in blue above the high of the bar. The down counts in the buy setup are shown in red below the low of the bar. Ensign's Design Your Own™ study was used to construct the indicator.

Study Name	Study Location	Marker Location	Label Location	Grid Tab
DeMark Count	Chart ▼	Study Value ▼	Margin 1 ▼	Default ▼

Study Scale			Variables File	Message Location
Chart Scale ▼			-DemarkCount ▼	None ▼

Message Text   Font   True   False   Panel

[ ] ■ ■ ■ ■

Sound

( ) Silent   ( ) Beep   ( ) Voice   ( ) WAV   [ ] Sound once per bar

WAV file

[ ] [Browse]

Trading System

Price

Close ▼   Quantity [1]

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Function ▼	4 Momentum ▼	= Momentum(#2, [#]) ▼	4 ▼	Close ▼	0 ▲▼
	▼	(▼	▼	▼	0 ▲▼)

A	Function	[Momentum] := Momentum(Close, 4)	[ ]		
B	Branching	if ([Momentum] > 0) then do next row	[ ]		
C	Expression	[Up Count] := [Up Count] + 1	[✓] BAR	■	Above High
D	Action	if ([Momentum] <= 0) then [Up Count] := 0	[ ]		
E			[ ]		
F	Branching	if ([Momentum] < 0) then do next row	[ ]		
G	Expression	[Down Count] := [Down Count] + 1	[✓] BAR	■	Below Low
H	Action	if ([Momentum] >= 0) then [Down Count] := 0	[ ]		

A – Calculate the 4 bar momentum (change) in the close of the bars.

B – Conditionally execute the next row to increment the Up Count variable.

C – Increment the Up Count variable and post its value above the bar's high.

D – Reset the Up Count variable to zero when the momentum is negative.

F – Conditionally execute the next row to increment the Down Count variable.

G – Increment the Down Count variable and post is value below the bar's low.

H – Reset the Down Count variable to zero when the momentum is positive.

DeMark Range Expansion Index

The DeMark REI is a market-timing oscillator described in DeMark on Day Trading Options, by T.R. DeMark and T.R. Demark, Jr., McGraw Hill, 1999.   The REI oscillator typically produces values of -100 to +100 with 45 or higher indicating overbought conditions and -45 or lower indicating oversold.
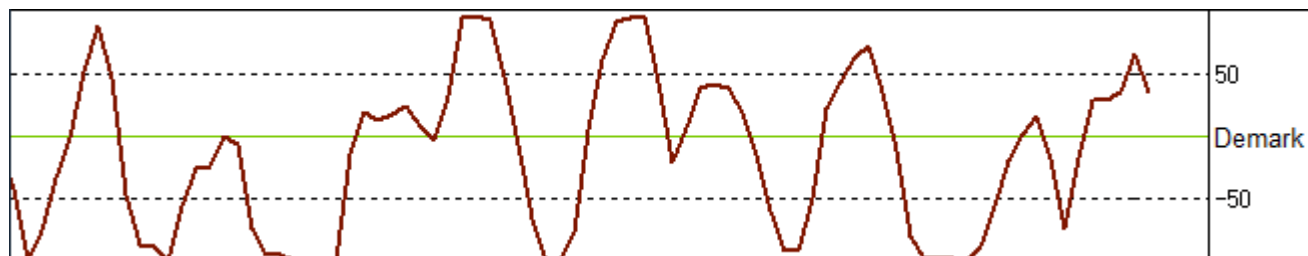
Here is how Tom DeMark describes the calculation of Range Expansion Index:

"The first step in calculating the REI is to add together the respective differences between the current day's high and the high two days earlier and the current day's low and the low two days earlier. These values will be positive or negative depending on whether the current day's high and low are greater or less than the high and low two days earlier. To prevent buying or selling prematurely into a steep price decline or advance, two additional conditions should be met to qualify a positive or negative value on a particular day: 1) either the high two days earlier must be greater than or equal to the close seven or eight days ago, or the current day's high must be greater than or equal to the low five or six days ago; 2) either the low two days earlier must be less than or equal to the close seven or eight days ago, or the current day's low must be less than or equal to the high five or six days ago. If either of these conditions are not satisfied, a zero value is assigned to that day. If they both are, the daily values (the differences between the highs and lows) are summed , and the specific value for that next day is determined. Next, all the positives and negative values are added together over a five-day period. This value is then divided by the absolute value price movement of each day over the five-day period. The numerator of the calculation can be either positive, negative or zero, because each day's value is summed for five days, but the denominator is always positive because it is only concerned with the differential price movement itself. This value is then multiplied by 100. Consequently, the REI can fluctuate between +100 and -100."

Formula

> HighDelta = High - High(-2)
> LowDelta = Low - Low(-2)
>
> Flag1 = High >= Low(-5) OR High >= Low(-6)
> Flag2 = High(-2) >= Close(-7) OR High(-2) >= Close(-8)
> Flag3 = Low <= High(-5) OR Low <= High(-6)
> Flag4 = Low(-2) <= Close(-7) OR Low(-2) <= Close(-8)
> Flag = (Flag1 OR Flag2) AND (Flag3 OR Flag4);
>
> IF Flag then Numerator = HighDelta+LowDelta else Numerator = 0
> Denominator = Abs(HighDelta) + Abs(LowDelta)
> REI = 100 * Sum(Numerator, 5) / Sum(Denominator, 5) ;

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Expression ▾ 192 Flag1	▾ = [$H]		▾ >= ▾ Low		▾ -5 ▴▾
彐 彐 ⟲	OR	▾ ([$H]	▾ >= ▾ Low		▾ -6 ▴▾)

	Category	Expression
A	Expression	[190 H2] := High[-2]
B	Expression	[191 L2] := Low[-2]
C	Expression	[192 Flag1] := ([$H] >= Low[-5]) OR ([$H] >= Low[-6])
D	Expression	[193 Flag2] := ([190 H2] >= Close[-7]) OR ([190 H2] >= Close[-8])
E	Expression	[194 Flag3] := ([$L] <= High[-5]) OR ([$L] <= High[-6])
F	Expression	[195 Flag4] := ([191 L2] <= Close[-7]) OR ([191 L2] <= Close[-8])
G	Expression	[196 Sum] := (High - [190 H2]) + (Low - [191 L2])
H	Function	[200 AbsH] := abs((High - [190 H2]))
I	Function	[197 AbsL] := abs((Low - [191 L2]))
J	Expression	[198 Flag] := ([192 Flag1] OR [193 Flag2]) AND ([194 Flag3] OR [195

A, B – Store the High and Low from 2 bars back in variables [190] and [191].

C, D, E, F – Implements the Flag1 through Flag4 expressions.

G – Variable [196] is the expression HighDelta + LowDelta.

H, I – Implements the Abs(HighDelta) and Abs(LowDelta).

J – Is the Flag formula.

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Action ▾ 199 Num	▾ = if ## then V := #2 else V := 0 ▾		▾ [196 Sum]		▾ 0 ▴▾
彐 彐 ⟲		▾ ([198 Flag]	▾	▾		▾ 0 ▴▾)

	Category	Expression
A	Action	if ([198 Flag]) then [199 Num] := [196 Sum] else [199 Num] := 0
B	Expression	[201 Den] := [200 AbsH] + [197 AbsL]
C	Function	[202 SumNum] := Sum([A], 5)
D	Function	[203 SumDen] := Sum([B], 5)
E	Expression	[204 TDREI] := (100 * [202 SumNum]) / [203 SumDen]

A – Implements the conditional assignment to the Numerator.

B – This calculates the Denominator variable.

C – Variable [202] is the Sum(Numerator, 5)

D – Variable [203] is the Sum(Denominator, 5)

E – Implements the REI formula and plots this study value in a sub-window.

DeMarker Study

The DeMarker study by Tom Demark is an attempt to overcome the shortcomings of classical overbought / oversold indicators. The DeMarker study identifies potential price bottoms and tops. It accomplishes this by making price comparisons from one bar to the next and measuring the level of price demand.

Formula

IF High > High(-1) then HighDelta = High – High(-1) else HighDelta = 0
IF Low < Low(-1) then LowDelta = Low(-1) – Low else LowDelta = 0
DeMarker = 100 * Sum(HighDelta, 13)/(Sum(LowDelta, 13) + Sum(HighDelta, 13));

C – The Sum function references the values on row A, and not the variable HighDelta. The variable has a single value, whereas, row A has a series of values that can be summed.

D – See the comment for row C.   The Sum function needs to reference row B values.

ESPL Script

For the sake of comparison and illustration, the DeMarker study will be implemented with an ESPL script.
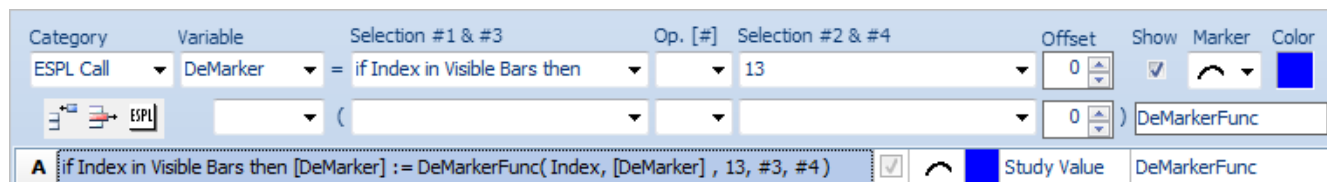
A – The function call will be made just for the visible bars to reduce the CPU calculation burden.   The parameter for the Sum function will be passed in the Selection #2 field.

This ESPL script implements the DeMarker study.

```
ESPL Editor
 6   function DeMarkerFunc(index, v, count, n3, n4);
 7   begin
 8      SumHigh := 0;
 9      SumLow  := 0;
10      for j := index - count + 1 to index do begin
11         if High(j) > High(j-1) then SumHigh := SumHigh + High(j)  - High(j-1);
12         if Low(j)  < Low(j-1)  then SumLow  := SumLow  + Low(j-1) - Low(j);
13      end;
14      if SumHigh+SumLow > 0 then Result := 100 * SumHigh / (SumHigh + SumLow)
15      else Result := 0;
16   end;
```
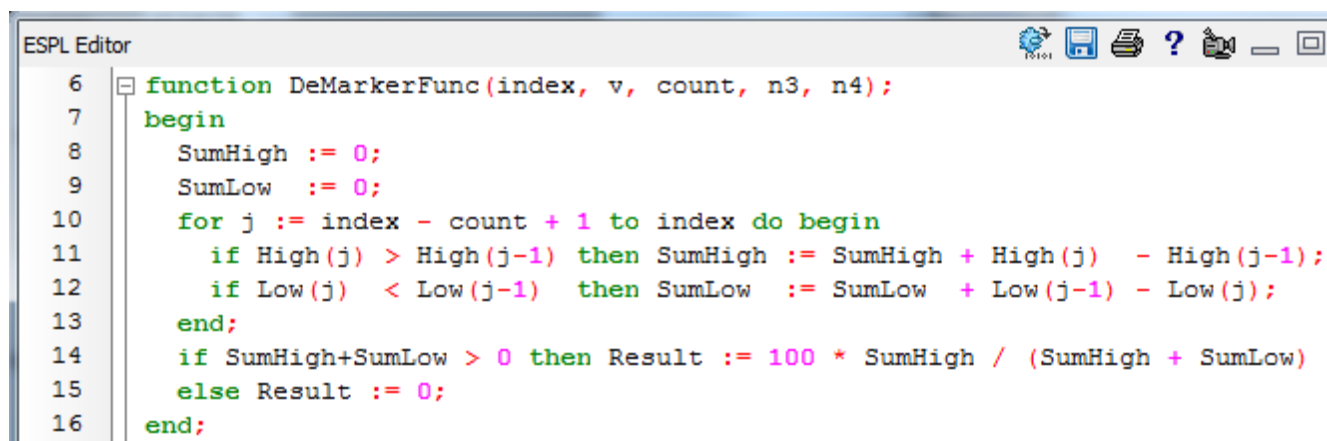
Index – This parameter is the bar index passed by the DYO for the bar being evaluated.

Count – This is the parameter passed by the DYO from the Selection #2 field, which is a 13.

High(index) - The High array holds the bar high values. (see line 11)
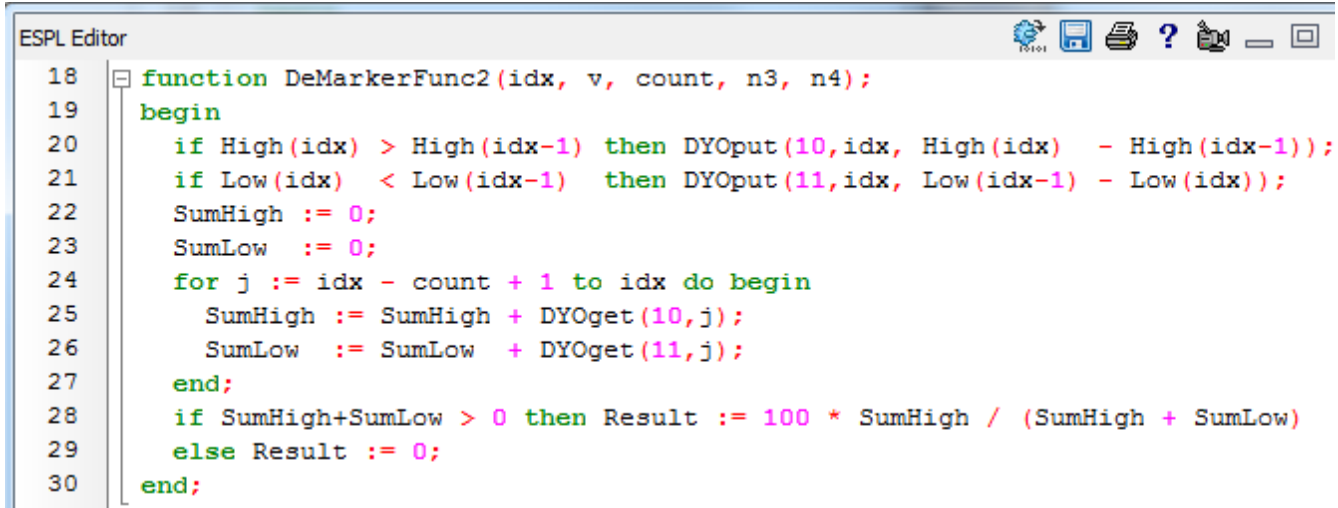
Low(index) - The Low array holds the bar low values. (see line 12)

Speed – The ESPL implementation, unfortunately, requires more CPU cycles to calculate. The DYO calculation time for 1650 bars was 0.01 seconds, whereas the ESPL calculation time for the same 1650 bars was 0.43 seconds. Four tenths of a second might not be noticed when opening a chart containing 1650 bars, but would be noticed if the ESPL calculation was performed on 20,000 bars.

Wisely choose an ESPL Call selection that will minimize the calculation burden. The example shown calculates for just the visible bars, which for a typical chart might be 65 bars though the chart file might contain thousands of bars.   The ESPL calculation time for 65 visible bars was 0.03 seconds.

DYO Arrays

This variation of the DeMarker script is more efficient by doing the bar calculations one time per call, and saving the calculations to DYO row arrays. (lines 20-21) The loop for the sums reads the values stored in the DYO arrays. (lines 24-27). The calculation time across 1650 bars dropped from 0.43 to 0.23 seconds.

```
ESPL Editor
18   function DeMarkerFunc2(idx, v, count, n3, n4);
19     begin
20       if High(idx) > High(idx-1) then DYOput(10,idx, High(idx)  - High(idx-1));
21       if Low(idx)  < Low(idx-1)  then DYOput(11,idx, Low(idx-1) - Low(idx));
22       SumHigh := 0;
23       SumLow  := 0;
24       for j := idx - count + 1 to idx do begin
25         SumHigh := SumHigh + DYOget(10,j);
26         SumLow  := SumLow  + DYOget(11,j);
27       end;
28       if SumHigh+SumLow > 0 then Result := 100 * SumHigh / (SumHigh + SumLow)
29       else Result := 0;
30     end;
```

DYOput(row, index, value) - This function writes the Value parameter into one of the DYO row arrays, 1 through 12, at the bar's index position. Row A is array 1, and row L is array 12. Since the example DYO only used row A, the other 11 arrays were available for general storage.

DYOget(row, index) - This function returns a Value from one of the DYO row arrays, 1 through 12.

It is a good programming practice to verify that the denominator is non-zero to avoid getting the error 'Invalid Float Operation' if a division by zero were to be performed. (lines 28-29)

A DYO design can be a combination of DYO statements and calls to ESPL functions. As demonstrated in this last example, ESPL has full access to read and write DYO row arrays. For example, the ESPL function could have written the row values used by the DYO function to Sum(#2, [#]), thus avoiding a time consuming loop in the ESPL function.

ESPL File

A single ESPL script file is used by all DYOs.   This ESPL script may contain multiple functions.

The file path and name is C:\Ensign 10\ESPL\DYO.psc. The file is automatically loaded for use by any DYO and is edited by clicking the ESPL button on the DYO property form. Edits are saved to this file.

The first button on the Editor toolbar is the Compile button. Click this button to verify the ESPL script syntax is correct and it will compile.

Dynamic Fibonacci Levels

This example adjusts the Fibonacci levels dynamically as the daily high/low range changes

		Selection #1 & #3		Selection #2 & #4		Show	Marker	Color
A	Expression	[0] := Today High		✓	⌐	Study Value	High	
B	Expression	[0] := Today Low		✓	⌐	Study Value	Low	
C	Function	[0] := Fib(1.618 * (Today High - Today Low) + Today Low)		✓	⌐	Study Value	1.618	
D	Function	[0] := Fib(1.272 * (Today High - Today Low) + Today Low)		✓	⌐	Study Value	1.272	
E	Function	[0] := Fib(0.618 * (Today High - Today Low) + Today Low)		✓	⌐	Study Value	0.618	
F	Function	[0] := Fib(0.5 * (Today High - Today Low) + Today Low)		✓	⌐	Study Value	0.5	
G	Function	[0] := Fib(0.382 * (Today High - Today Low) + Today Low)		✓	⌐	Study Value	0.382	
H	Function	[0] := Fib(1.272 * (Today Low - Today High) + Today High)		✓	⌐	Study Value	1.272	
I	Function	[0] := Fib(1.618 * (Today Low - Today High) + Today High)		✓	⌐	Study Value	1.618	

A – Plot the daily High.

B – Plot the daily Low.

C - I – Calculate the Fibonacci level using the Fib function.   The level value is entered as the #2 selection. Note that the #3 and #4 selections are reversed on row H and I to calculate extensions below the daily Low.

Dynamic Support and Resistance

Three chart prices are used to calculate Support and Resistance lines. Typically the High, Low, and Close prices of a daily range are used. However, this DYO example will use 3 different prices which are related to the 1st session open. The 3 prices will be the session open price, and the high and low of the 1st hour.

Formula

Pivot   = (High Price + Low Price + Open Price) / 3
Range = High Price – Low Price

4R [4th Resistance] = 2R + Range
3R [3rd Resistance] = 1R + Range
2R [2nd Resistance] = Pivot + Range
1R [1st Resistance] = Pivot + (Pivot - Low)
H   [1st Hour High]
P   [Pivot Price]
L   [1st Hour Low]
1S [1st Support] = Pivot - (High-Pivot)
2S [2nd Support] = Pivot - Range
3S [3rd Support] = 1S – Range
4S [4th Support] = 2S – Range
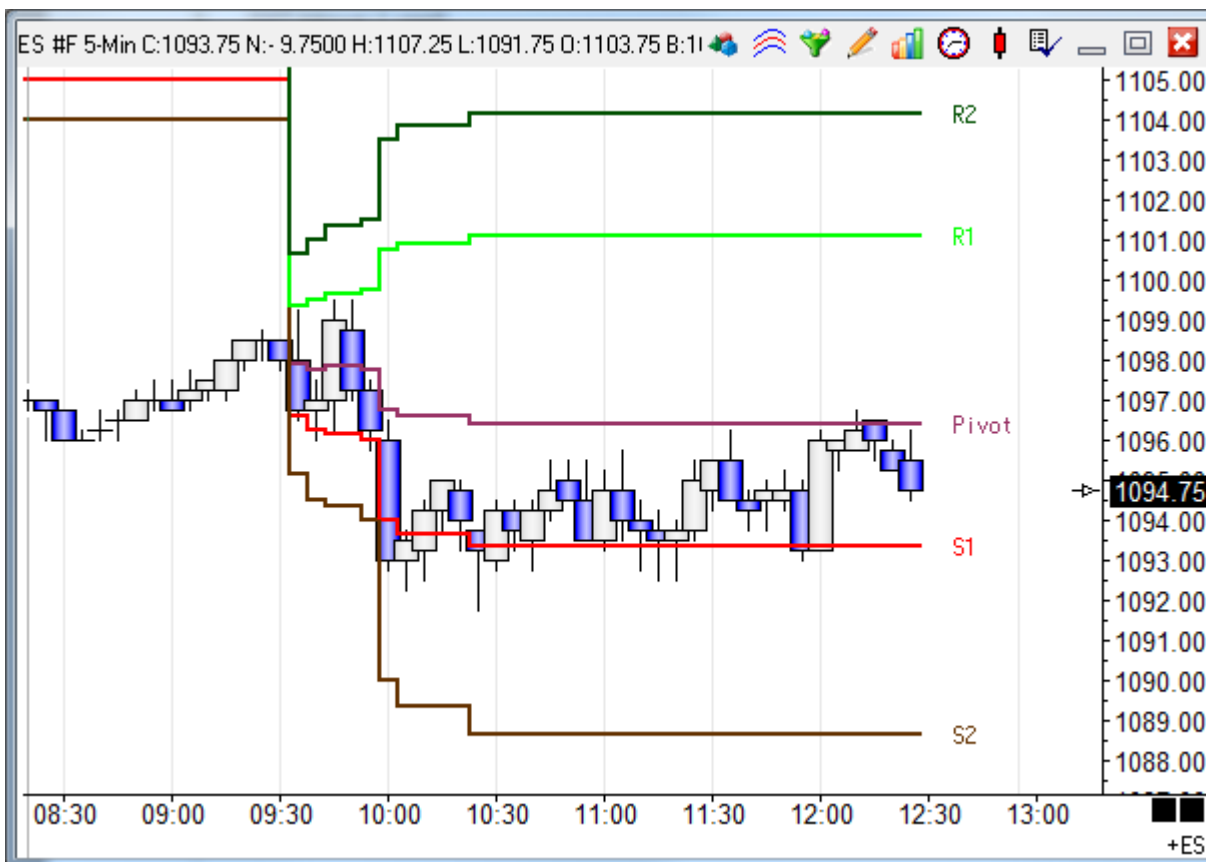
	Study Name	Study Location	Marker Location	Label Location	Grid Tab	
	Dynamic S & R Lines	Chart ▾	Study Value ▾	Margin 2 ▾	Default ▾	☐ U:
	Study Scale			Variables File	Message Location	☐ Pl
	Chart Scale ▾			-DynamicS&R ▾	None ▾	☐ Pr
						☐ Cl
						☐ Sl

Message Text   Font True False Panel
☐ Er
☐ Al
✔ A(

Sound
◉ Silent   ○ Beep   ○ Voice   ○ WAV   ☐ Sound once per bar

Trading System
Price   Quantity
Close ▾   1 ▲▼

WAV file
[Browse]

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show
Chart Value ▾	58 High ▾	= #2 Minute High after 1st Op ▾	▾	60 ▾	0 ▲▼	☐
≡ ⇥ ⇤	▾	(▾	▾	▾	0 ▲▼)	

A	Chart Value	[High] := 60 Minute High after 1st Open	☐				
B	Chart Value	[Low] := 60 Minute Low after 1st Open	☐				
C	Expression	[Open] := Session 1 Open	☐				
D	Function	[Pivot] := Sum([Open], 3) / 3	✔	⌐		Study Value	Pivot
E	Expression	[S1] := (2 * [Pivot]) - [High]	✔	⌐		Study Value	S1
F	Expression	[S2] := [Pivot] - ([High] - [Low])	✔	⌐		Study Value	S2
G	Expression	[R1] := (2 * [Pivot]) - [Low]	✔	⌐		Study Value	R1
H	Expression	[R2] := [Pivot] + ([High] - [Low])	✔	⌐		Study Value	R2

A – Save the high of the 1st hour following the 1st session open in a variable named High. The duration of the period is the #2 selection which is 60 minutes in this example.

B – Save the low of the 1st hour following the 1st session open in a variable named Low.

C – Save the open of the 1st session in a variable named Open.

D – Calculate the Pivot by averaging [Open], [High], and [Low]. The 3 variables used are adjacent to each other with [Open] being first on the variable list. This is an important relationship because the function averages adjacent variables with the variable which is 1st in the list being named in the function parameter. The quantity of 3 comes from the Op. [#] field.

E – Calculate 1st Support.

F – Calculate 2nd Support.

G – Calculate 1st Resistance.

H – Calculate 2nd Resistance.

The line labels use Label Location Column 3 so they show in the right side margin.

Fractal Dimension
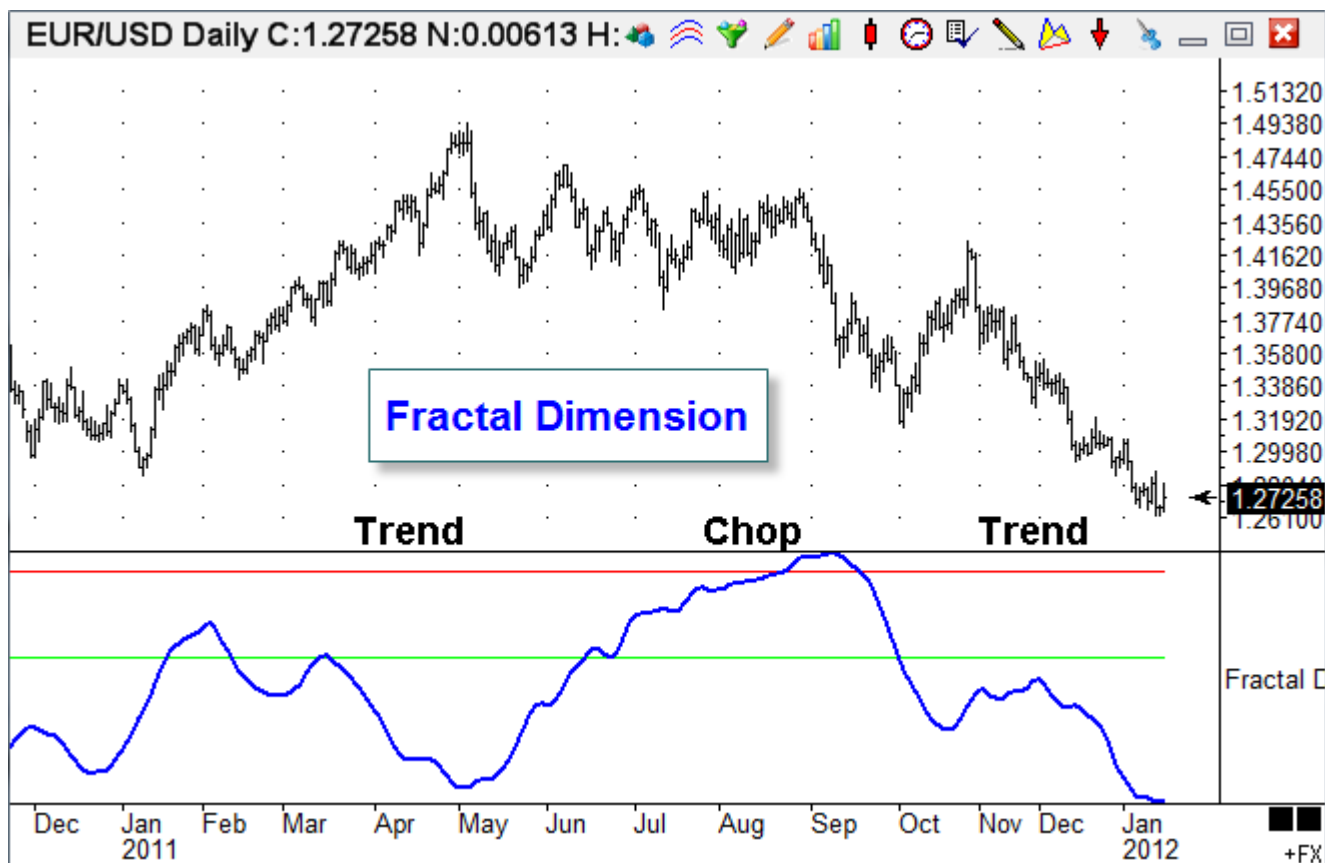
Fractal Dimension is a tool to help differentiate whether a market is trending or cycling. See the Fractal Dimension article in the *Stocks & Commodities* magazine, June 2010, by John Ehlers and Ric Way. Three DYOs are used in the Fractal Dimension template to implement the formula.

Part 1:

A	Expression	[N] := 30
B	Expression	[Price] := Close
C	Expression	[Smooth] := (2 * [B][-1]) + (2 * [B][-2])
D	Expression	[Smooth] := ([B] + [Smooth]) + [B][-3]
E	Expression	[Smooth] := [Smooth] / 6
F	Expression	[HalfN] := [N] / 2
G	Expression	[MinusN2] := - [HalfN]
H	Expression	[SmoothN] := [E][-15]
I		
J	Function	[HH] := Highest([H], [HalfN])
K	Function	[LL] := Lowest([H], [HalfN])
L	Expression	[N2] := ([HH] - [LL]) / [HalfN]

A – Define a variable N with the period parameter. The period parameter must be an even number, such as 30.   The variable index is [230].

B – Define the price variable that is to be used as the data point for the formula.

C, D, E – Implement the formula: Smooth = (Price + 2*Price[-1] + 2*Price[-2] + Price[-3]) / 6

F – Define a variable that is half the period parameter. The variable index is [229].

G – Define a variable that is a negative half period. This parameter is used as the offset on Row H.   The variable index is [228].

H – Defines a new stream of Prices that are derived from Row E but shifted rightward by a half period.   The Offset is entered as 228 so that variable [228]'s value is used.

J, I – Find the Highest High and Lowest Low of the Row H data set which was shifted.

L – Calculate the midpoint of the range's spread.

Part 2:

A	Expression	[0] := [Smooth]
B	Function	[HH] := Highest([A], [HalfN])
C	Function	[LL] := Lowest([A], [HalfN])
D	Expression	[N1] := ([HH] - [LL]) / [HalfN]
E		
F	Function	[HH] := Highest([A], [N])
G	Function	[LL] := Lowest([A], [N])
H	Expression	[N3] := ([HH] - [LL]) / [N]
I		
J	Function	[LogN1N2] := log10(([N1] + [N2]))
K	Function	[LogN3] := log10([N3])
L	Function	[Log2] := log10(2)

A – Copy the Smooth data from Part 1. Row A will have the original unshifted Smooth data set. The Highest and Lowest functions need to reference a row of data and not the variable named Smooth.

B, C – Find the Highest High and Lowest Low of the Row A data set.

D – Calculate the midpoint of the range's spread. The data set size is a half period.

F, G – Find the Highest High and Lowest Low of the Row A data set.

H – Calculate the midpoint of the range's spread. The data set size is a full period.

J, K, L – Define variables holding log10 values needed by the next DYO.

Part 3:

	Study Name	Study Location	Marker Location	Label Location	Grid Tab
	Fractal Dim, part 3	Window 1	Study Value	None	None

Study Scale		Variables File	Message Location
Data Set		-FractalDim	None

Message Text   Font   True False Panel

Sound
○ Silent   ○ Beep   ○ Voice   ○ WAV   ☐ Sound once per bar

Trading System
Price   Quantity
Close   1

WAV file   Browse

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Expression ▼	18 Ratio ▼ =	[Ratio] ▼	/ ▼	2 ▼	0
	+ ▼	([H] ▼	/ ▼	2 ▼	-1

A	Action	Row [D] := (1.5)	☐		
B	Branching	if (Bar Index <= [N]) then abort execution	☐		
C	Expression	[Ratio] := ([LogN1N2] - [LogN3]) / [Log2]	☐		
D	Expression	[Ratio] := ([Ratio] / 2) + ([H][-1] / 2)	☐		
E			☐		
F	Expression	[0] := 1.6	☑	⌐ 🟥	Study Value
G	Expression	[0] := 1.4	☑	⌐ 🟩	Study Value
H	Function	[Fractal] := Simple Average([D], 20)	☑	⌒ 🟦	Study Value
I			☐		
J	Expression	[0] := [Fractal] > 1.6	☐	ZON 🟫	
K	Expression	[0] := ([Fractal] > 1.4) AND ([Fractal] <= 1.6)	☐	ZON 🟧	
L	Expression	[0] := [Fractal] <= 1.4	☐	ZON 🟩	

A, B – Initialize the Row D values to 1.5 until N bars are on the chart.   Abort further calculation when the bar index is less than the N period parameter.

C, D – Calculate: Ratio := ((log(N1 + N2) – log(N3)) / log(2) + Fractal[-1]) / 2

F, G – Optionally plot scale lines at 1.4 and 1.6.

H – Calculate and plot: Fractal := Average(Ratio, 20)

J, K, L – Optionally plot colored zones based on the level of the Fractal Dimension.

Suggested use: Have trading system signals be based on trending tools when the Fractal Dimension is below 1.4, and use signals from overbought and oversold tools when the Fractal Dimension is above 1.4.

HiLo Activator

This template implements the Gann HiLo Activator.   It's a three-period moving average of the highs or lows plotted one day forward (see page 55 of the February 1998 S&C). If the market is in an uptrend and today is day 4, the hi-lo activator for day 4's trading is the low from days 1, 2 and 3 added together and divided by three. This value is the hi-lo activator for all of day 4. It does not change until the close of day 4. Then, for day 5, the hi-lo activator is the low from days 2, 3 and 4, added together and divided by three. The same concept applies for calculating the highs.
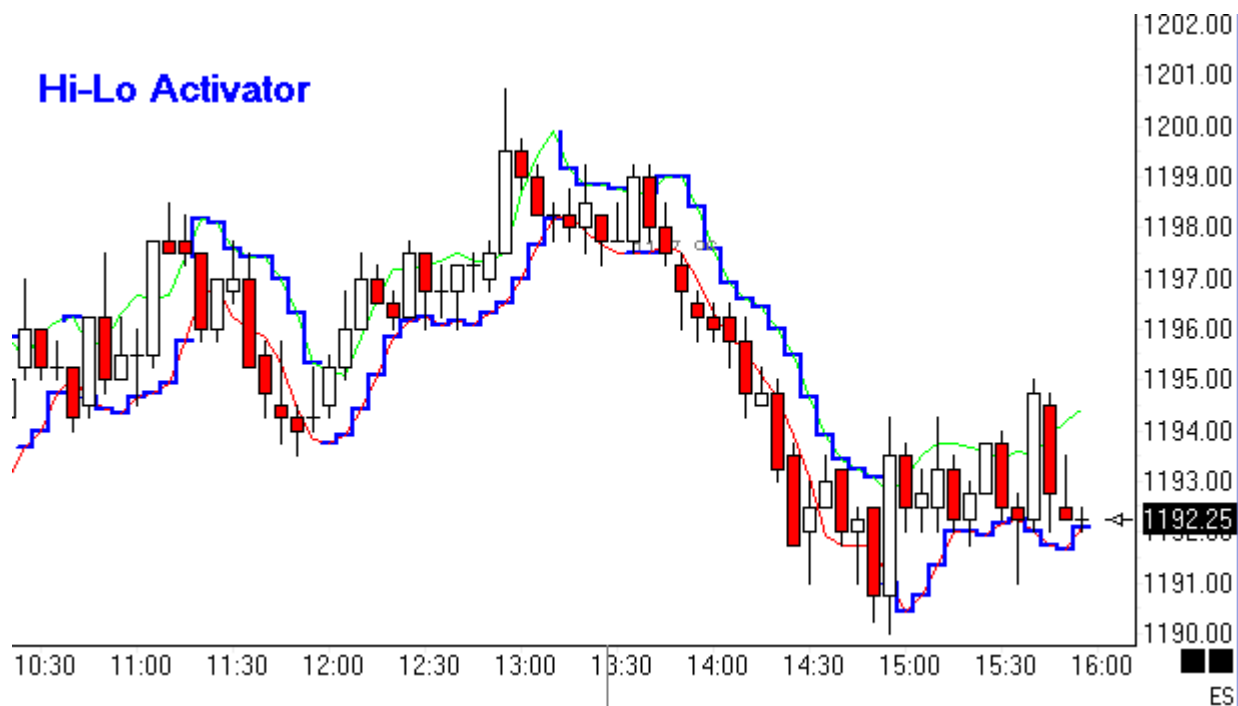
A	Function	[10 AveHigh] := Simple Average(High[-1], 3)	☐ ⌒		🟩
B	Function	[11 AveLow] := Simple Average(Low[-1], 3)	☐ ⌒		🟥
C	Expression	[0] := [10 AveHigh] < Close	☐		
D	Expression	[0] := [11 AveLow] > Close	☐		
E	Function	[1 HighCount] := Bars since True([C])	☐		
F	Function	[2 LowCount] := Bars since True([D])	☐		
G	Expression	[0] := [1 HighCount] > [2 LowCount]	☑ ⌐	🟦	[A] Value
H	Expression	[0] := [1 HighCount] < [2 LowCount]	☑ ⌐	🟦	[B] Value

A, B – Calculate the 3 period averages. Note the bar offset of -1 to exclude today.

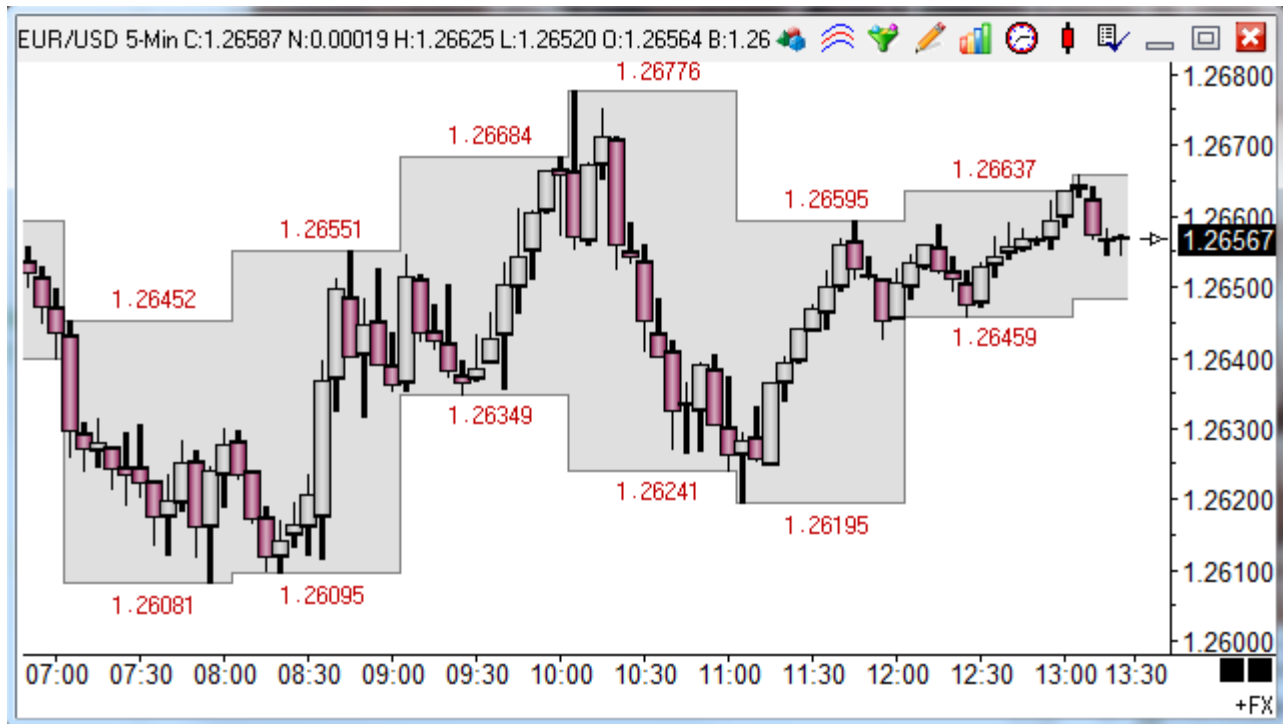C, D – Test to see when the averages are below or above the Close.

E, F – Count how many bars the tests on rows C and D have been True.

G – When in a Down Trend, plot the Simple Average of the Highs from row A.

H – When in an Up Trend, plot the Simple Average of the Lows from row B.

Hourly Bars

This example groups intra-day chart bars into hourly groups and labels the group with its 60-minute high and low.

Study Name	Study Location	Marker Location	Label Location	Grid Tab	
Hourly Bars	Chart	Study Value	None	Default	☐ Use as Default
					☑ Draw Behind Bars
Study Scale			Variables File	Message Location	☐ Privatize
					☐ Close Only
Chart Scale			-HourlyBars	None	☐ Show Values
Message Text		Font True False Panel			☐ Email
		■ ■ ■ ■			☐ Auto Remove
					☑ Auto Completion

Sound
◉ Silent ◯ Beep ◯ Voice ◯ WAV ☐ Sound once per bar

Trading System
Price — Last   Quantity 1   Commission 0

WAV file ___ Browse

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
Chart Value	61 High	= #2 Minute High		60	0	☑	⬤	
		(			0			)

A	Chart Value	[High] := 60 Minute High	☑	⬤		Study Value
B	Chart Value	[Low] := 60 Minute Low	☑	⌐		Study Value
C	Expression	[0] := [High]	☑	⌐		Study Value
D	Expression	[Minutes] := (Minutes from Midnight / 60) - 0.55	☐			
E	Function	[0] := trunc([Minutes])	☐			
F	Flag	[TurnsUp] := [E] turns up	☐			
G	Branching	if Not [TurnsUp] then abort execution	☐			
H	Expression	[0] := [High] + 0.0003	☑	LBL		Study Value [A]
I	Expression	[0] := [Low] - 0.0003	☑	LBL		Study Value [B]

A - B – The high and low for a 60-minute period are returned. The spread fill marker colors the space between these 2 prices. The marker uses 2 values and these values come from the row with the marker and the next DYO row.

C – This row is needed for the sake of drawing the line on the high value from row A.   The marker on row A was used for the spread fill coloring.

Row D, E, F and G accomplish limiting the values to show once per 60 min period instead of on every bar.

E – The value subtracted on row E shifts the position of the answer shown either to the right or to the left.

H - I – The values on rows H and I accomplish a vertical offset so the High and Low values are not printed on top of the row B and row C lines.   The LBL label marker uses the text in the label field as the marker, which is a value read from row A or row B.

Ichimoku

Formula

SL = (HHV(H, 26) + LLV(L, 26))/2;	kijun-sen line
TL = (HHV(H, 9) + LLV(L, 9))/2;	tenkan-sen line
DL = Ref(C, -25);	chikou span
Span1 = Ref((SL + TL)/2, -25);	senkou span A
Span2 = Ref((HHV(H, 52) + LLV(L, 52))/2, -25);	senkou span B

A, B, C – Plot the SL formula which averages the 26 bar Highest High and Lowest Low.

D, E, F – Plot the TL formula which averages the 9 bar Highest High and Lowest Low.

G – Plot the DL line which is the Close price from 25 bars back.

H – Plot the Span1 line which is the average of SL+TL, shifted forward 25 bars.

I, J, K – Plot the Span2 line which is the 52 bar HH and LL, averaged and shifted 25 bars.

L – Plot a line through the Close prices.

Example Ichimoku chart:
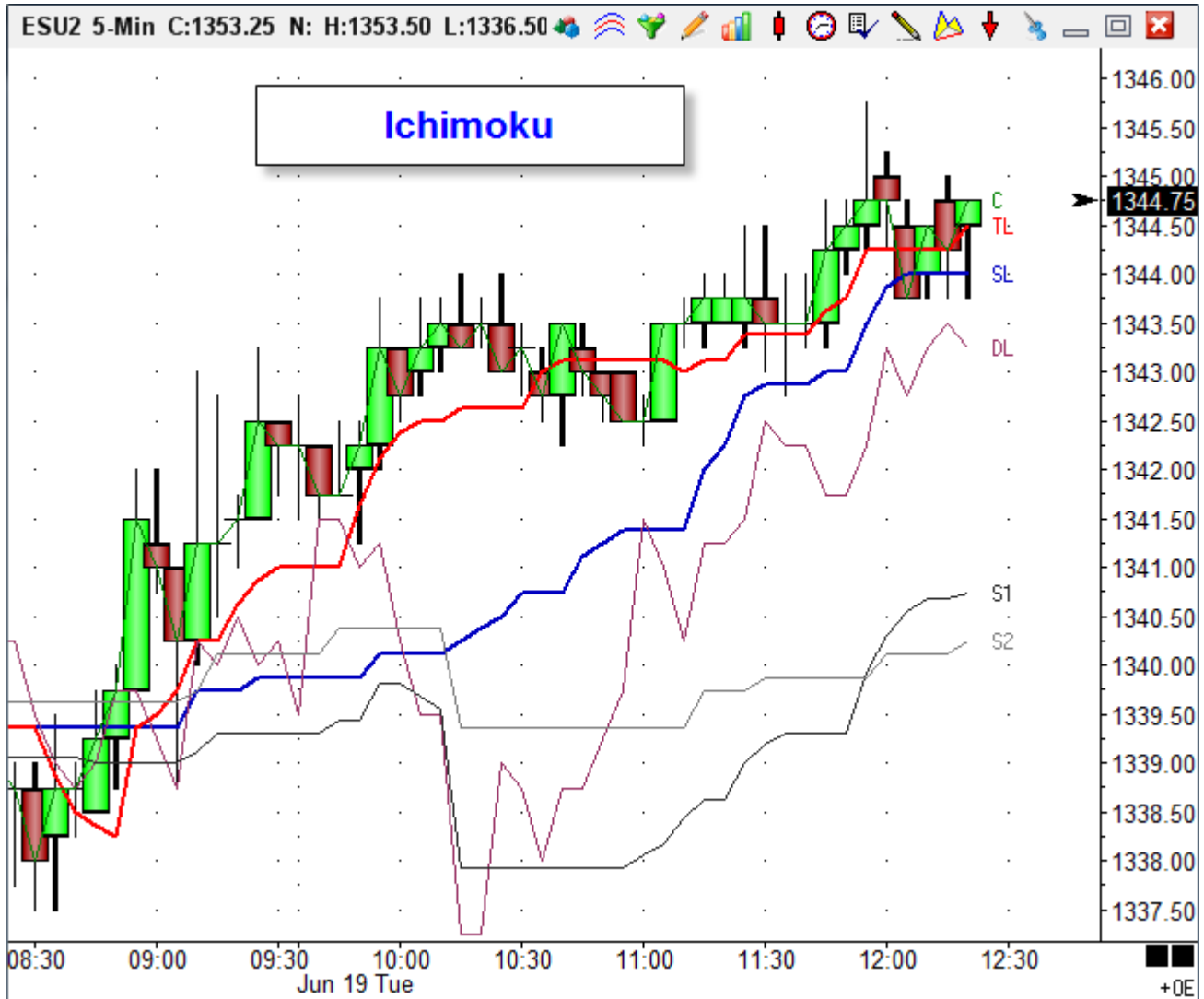
McClellan Summation Index

The McClellan Summation Index is this formula on the $ADD chart.

Formula

Summation Index = 1000 + (10%Trend – 5%Trend) – ((10 x 10%Trend) + (20 x 5%Trend))

> where:

> 5%Trend = 39-day EMA of (Advancers-Decliners)

> 10%Trend = 19-day EMA of (Advancers-Decliners)

The symbol for the Advancers-Decliners is delayed but is available through IB as $ADD.
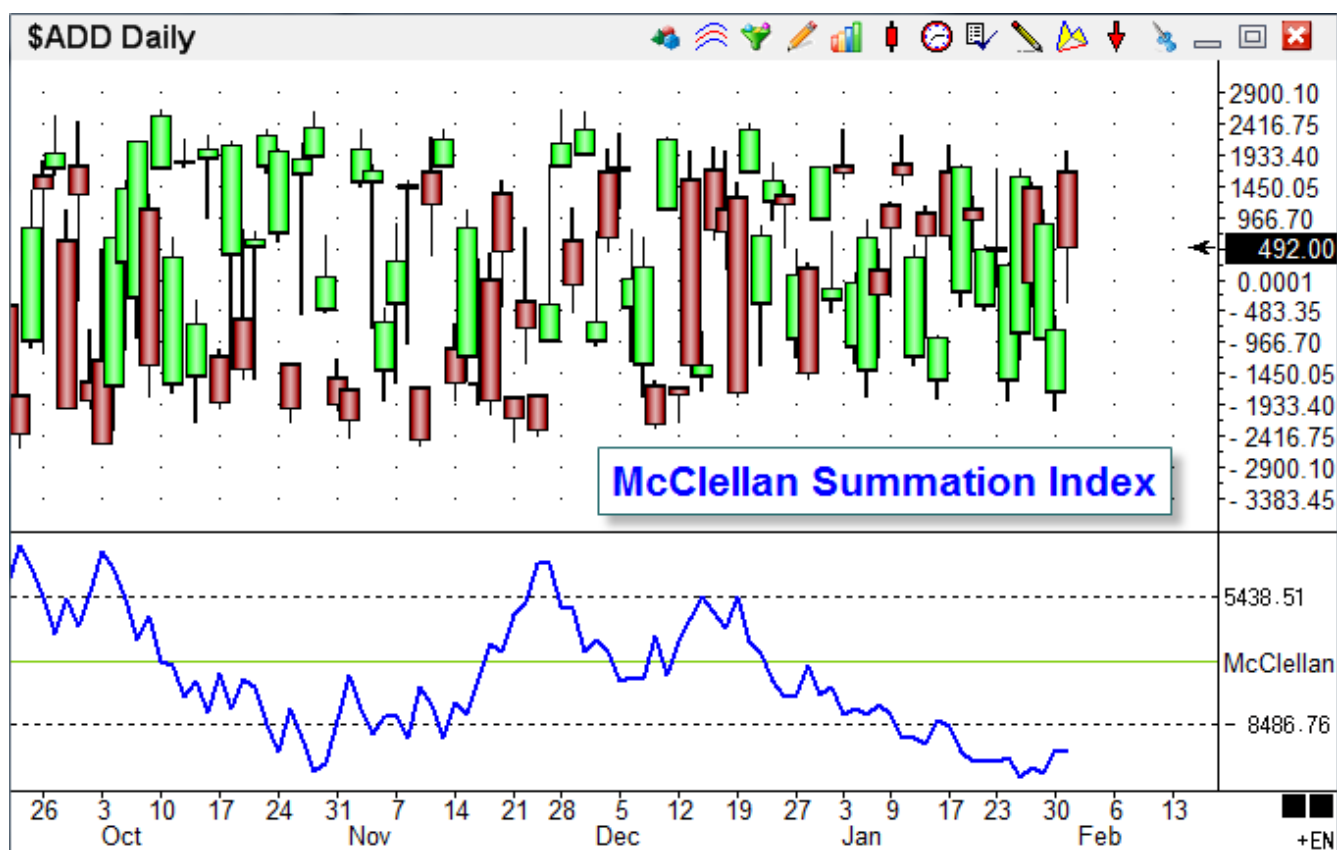
The formula is implemented with this DYO, and available on the Ensign web site using the Ensign 10 Package feature to download it.

	Study Name	Study Location	Marker Location	Label Location	Grid Tab

Study Name: McClellan Summation
Study Location: Window 1
Marker Location: Study Value
Label Location: None
Grid Tab: Default

Study Scale: Data Set +5%
Variables File: -McClellanSum
Message Location: None

Message Text: [ ] Font ■ (black) True █ (green) False █ (red) Panel █ (yellow)

Sound
- ◉ Silent ○ Beep ○ Voice ○ WAV ☐ Sound once per bar

WAV file: [ ] Browse

Trading System
- Price: Close
- Quantity: 1

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Function	2 5%Trend	= Exponential Ave(#2, [#])	39	Close	0

A	Function	[5%Trend] := Exponential Ave(Close, 39)	☐			
B	Function	[10%Trend] := Exponential Ave(Close, 19)	☐			
C	Expression	[0] := 1000 + ([10%Trend] - [5%Trend])	☐			
D	Expression	[0] := (10 * [10%Trend]) + (20 * [5%Trend])	☐			
E	Expression	[0] := [C] - [D]	☑	⌒	█	Study Value

A – Create the 39 period average.

B – Create the 19 period average.

C – Calculate the first half of the formula: 1000 + (10%Trend – 5%Trend)

D – Calculate the 2nd half of the formula: 10 * 10%Trend + 20 * 5%Trend

E – Subtract the two halves. Plot this result which is the Summation Index.

Move Alan Square

A DYO can relocate draw tools, such as Fibonacci Levels, Gann Square, Draw Lines, and Alan Square.   This example will relocate 2 Alan Squares so they automatically move forward each day to be positioned on the day just completed and on the day in progress. The two prices used in the relocation will be the High and Low from the prior day.
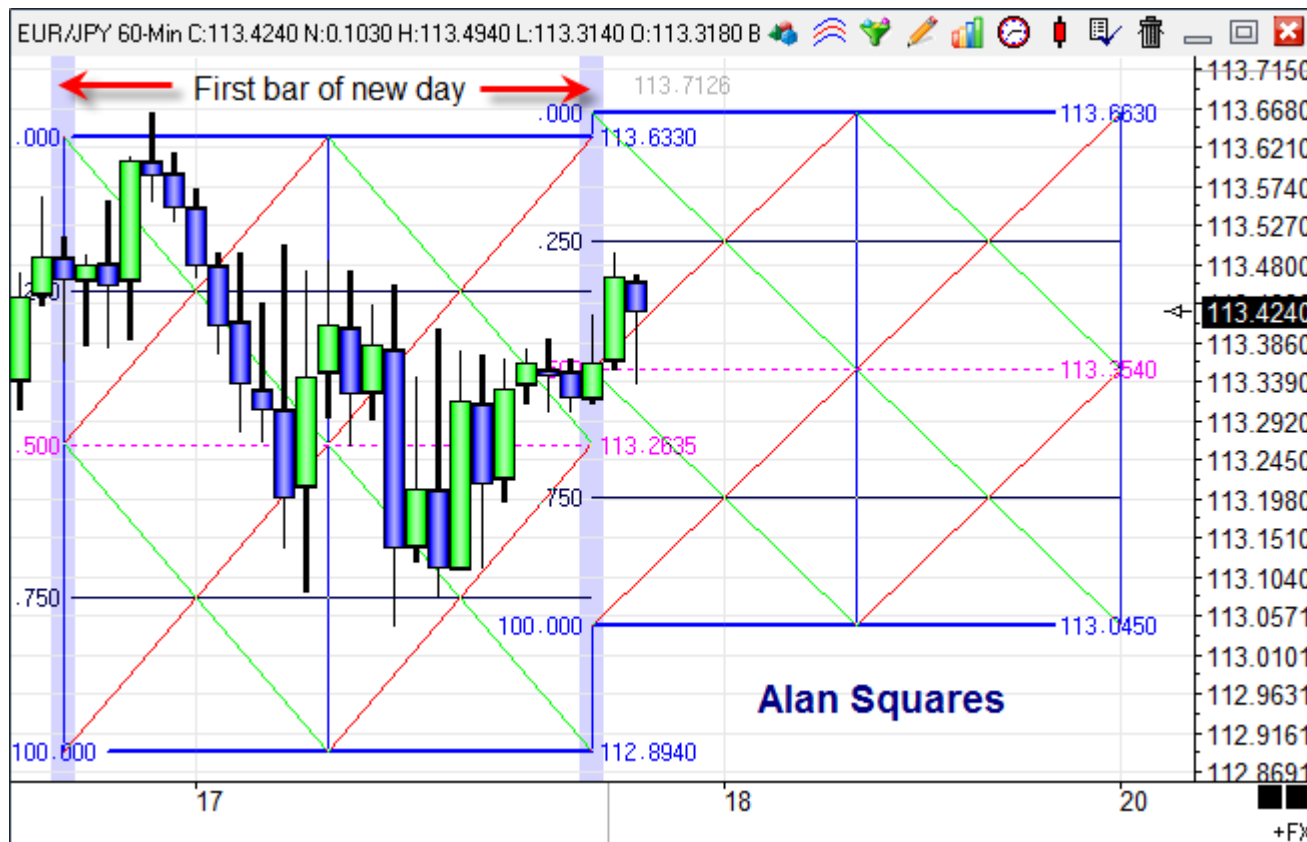
The chart has 2 Alan Square draw tools, and 2 DYOs.   The DYO that manages the placement of the Alan Square on the right will be discussed first.

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	
Action	0	= object(#2).Set Point A(#3,		[Handle]	
		(Bar Index		Yesterday High	

	Category			ZON	
A	Flag	[0] := 1st bar of Session 2		☐ ZON	
B	Expression	[0] := Yesterday High		☐	
C	Branching	if Not [A] then abort execution		☐	
D	Expression	[EndIndex] := Bar Index + 24		☐	
E	Study	[Handle] := Alan.Handle		☐	
F	Action	object([Handle]).Set Point A(Bar Index, Yesterday High)		☐	
G	Action	object([Handle]).Set Point B([EndIndex], Yesterday Low)		☐	

41

A – This flag is the trigger event for when to relocate the Alan Square.

B – Row B expression does not appear to be used by the DYO, but it is necessary to force the chart to maintain Yesterday values. Otherwise, lacking such a reference, the chart in its efficiency would skip calculating and maintaining daily values for Yesterday and Today.

C – This branching statement aborts further execution when evaluation is not on a trigger bar.

D – The row calculates the Index for the 2nd construction point for the Alan Square. This ending location is 24 bars into the future on this hourly chart which has 24 bars per day. If the example is used on some other time frame chart, the number of bars between the ending point and the beginning point will need to be used in the expression.

E – The Alan Square was selected in the drop down selection list for studies in the Selection #1 field. The Handle selection was picked form the bottom of the Selection #2 list box. The handle is a pointer to the chart object and exposes access to the object's properties. The handle returned was assigned to a Variable named [Handle].

F- This statement from the bottom of the Action selection list sets the properties for the 1st construction point for the Alan Square. The object handle is the Selection #2 parameter. The Selection #3 field passes the Index for locating the point horizontally on the chart. The Selection #4 field passes the Price for locating the point vertically on the chart. The Bar Index parameter is the index of the bar being evaluated, and would be the index of the trigger event which is the index of the 1st bar of the 2nd session. Yesterday's High is used as the price.

G – This statement is similar to Row F. The 2nd construction point for the Alan Square is set to the ending index calculated on Row D, and Yesterday's Low is the price.

Insufficient lines remain in this DYO to implement locating the 2nd Alan Square, so another DYO is used, which will be discussed next.

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4
Action ▼	244 ResetFlag ▼	= if ## then V := #2 ▼	▼	False ▼
☰ �™ ➙	Not ▼	([A] ▼	▼	▼

A	Flag	[0] := 1st bar of Session 2	☑	ZON	Stuc
B	Action	if (Not [A]) then [ResetFlag] := False	☐		
C	Branching	if Not [A] then abort execution	☐		
D	Branching	if [ResetFlag] then abort execution	☐		
E	Study	[Handle] := Alan.Handle	☐		
F	Action	object([Handle]).Set Point A([PriorIndex], [PriorYH])	☐		
G	Action	object([Handle]).Set Point B(Bar Index, [PriorYL])	☐		
H	Expression	[ResetFlag] := True	☐		
I	Expression	[PriorIndex] := Bar Index	☐		
J	Expression	[PriorYH] := Yesterday High	☐		
K	Expression	[PriorYL] := Yesterday Low	☐		

Row A and C are the same as discussed in the 1st DYO.

The chart maintains daily value for Today and Yesterday but not for the day before Yesterday. Therefore, this DYO will capture the values for Yesterday before they are replaced at the start of a new session. These old values for Yesterday High and Yesterday Low will be remembered in variables named [PriorYH] and [PriorYL].

Rows B, D and H work together to manage a ResetFlag so that the transfer of Yesterday's High and Low into the [PriorYH] and [PriorYL] happen a single time. If subsequent executions were not blocked, the current YH and YL would be transferred and the PriorYH and PriorYL would lose their values of the day before Yesterday.

B – Reset the flag to False when the chart is on any bar following the trigger bar.

D – Abort execution on a subsequent tick update. The Alan Square has already been moved on the first tick of the new session, and the YH and YL values were already moved.

H – Set the flag on the pass through the DYO that moves the Alan Square and updates the PriorYH and PriorYL values. By setting the [ResetFlag] to True, subsequent calls will be blocked.

E – Find the handle to the Alan Square object. Though it is not apparent from the script, the 2nd Alan Square was selected in the study drop down list. This the handle points to a different object than used in the 1st DYO.

F - G – These rows are similar to the discussion in the 1st DYO. [PriorIndex] is one of the variables remembered by this DYO, and is the index of the session bar 1 day earlier.

I - J - K – These rows update the Variables that are used to remember the values for the day ahead of Yesterday. Yesterday values are captured before they are changed to the new day.

A DYO can be used to read or write other properties from study and draw tool objects. These selections are in the Action category.

```
object( #2 ).Set Point A( #3, #4 )
object( #2 ).Set Point B( #3, #4 )
object( #2 ).Set Point C( #3, #4 )
object( #2 ).Set Param 1( #3 )
object( #2 ).Set Param 2( #3 )
object( #2 ).Set Param 3( #3 )
object( #2 ).Set Param U/D( #3 )
object( #2 ).Set Param L/R( #3 )
object( #2 ).Set Field( #3, #4 )
object( #2 ).Get Field( #3 )
```

Set Point A, B and C take 2 parameters, an Index passed in #3 and a Price passed in #4.

Set Param 1, 2, 3, U/D, and L/R have one parameter and that is the value to assign. Parameter 1 for a Moving Average study would be the 1st average parameter on the study property form. U/D is the up / down price adjustment. L/R is the horizontal offset adjustment.

Set Field can assign any of the object's properties. See the GetStudy documentation in the ESPL manual for a list of selection values to use in field #3. #4 is the value to assign.

Get Field can read any of the object's properties. See the GetStudy documentation in the ESPL manual for the selection values to pass in the #3 field.

Move Andrews Pitchfork

This example uses swing points from the Pesavento Patterns study to automatically position the Andrews Pitchfork, which has three construction points. The chart contains an Andrews Pitchfork, a Pesavento Patterns study, and a DYO.

	Category	Variable	Selection #1 & #3		Op. [#]	Selection #2 & #4	
	Study	0	= Pesavento g			Swing bar	
			(				

A	Study	[0] := Pesavento.Swing bar	
B	Branching	if Not [A] then abort execution	
C	Study	[PointAprice] := Pesavento.Swing 4 value	
D	Study	[PointBprice] := Pesavento.Swing 3 value	
E	Study	[PointCprice] := Pesavento.Swing 2 value	
F	Study	[PointAindex] := Pesavento.Swing 4 index	
G	Study	[PointBindex] := Pesavento.Swing 3 index	
H	Study	[PointCindex] := Pesavento.Swing 2 index	
I	Study	[Handle] := Andrews.Handle	
J	Action	object([Handle]).Set Point A([PointAindex], [PointAprice])	
K	Action	object([Handle]).Set Point B([PointBindex], [PointBprice])	
L	Action	object([Handle]).Set Point C([PointCindex], [PointCprice])	

The DYO reads 3 swing points from the Pesavento and sets the Andrews Pitchfork points.

Move Fibonacci Levels

This example is similar to the Move Alan Square example. A DYO will move a Fibonacci Levels draw tool to be positioned on a recent Pesavento swing. The chart contains a Fibonacci Levels draw tool, a Pesavento Patterns study, and a DYO.
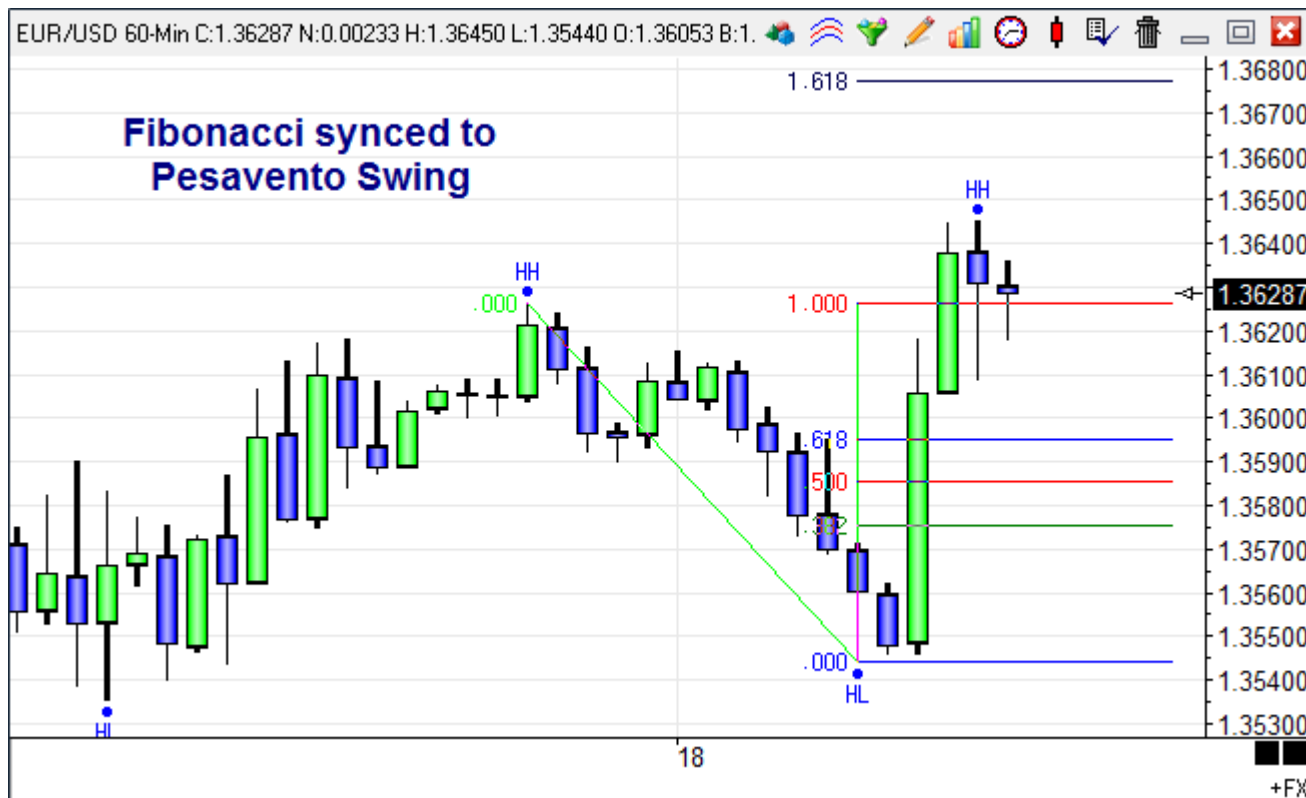
Category		Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	
Action	▼	0	▼ = object(#2).Set Point A(#3, ▼	▼	[Handle]	▼
			▼ ([PointAindex] ▼	▼	[PointAprice]	▼

				ZON	
A	Study	[0] := Pesavento.Swing bar		☐	ZON
B	Branching	if Not [A] then abort execution		☐	
C	Study	[PointAprice] := Pesavento.Swing 3 value		☐	
D	Study	[PointBprice] := Pesavento.Swing 2 value		☐	
E	Study	[PointAindex] := Pesavento.Swing 3 index		☐	
F	Study	[PointBindex] := Pesavento.Swing 2 index		☐	
G	Study	[Handle] := Fibonacci.Handle		☐	
H	Action	object([Handle]).Set Point A([PointAindex], [PointAprice])		☐	
I	Action	object([Handle]).Set Point B([PointBindex], [PointBprice])		☐	

C-F – These rows read the prices and bar indexes of the prior Pesavento swing.

G-I – These rows find the Fibonacci tool and set its two construction points.

Move Gann Square

This is another example of a DYO relocating a draw tool based on the swing points from the Pesavento Patterns study. A DYO will move a Gann Square to be positioned on a recent Pesavento swing. The chart contains a Gann Square draw tool, a Pesavento Patterns study, and a DYO.
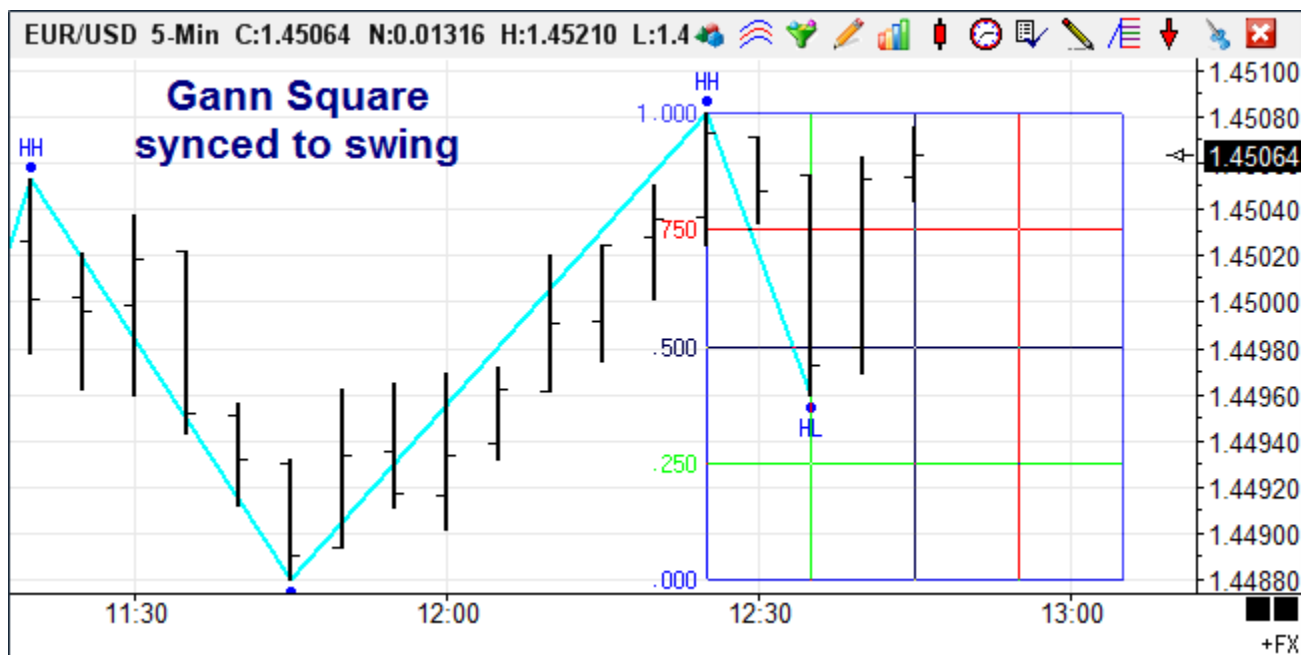
Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Action ▼	0 ▼	= object(#2).Set Point B(#3, ▼	▼	[15 Handle] ▼	0
⇉ ⇉ ⇶	▼	([14 Square B Index] ▼	▼	[10 B Price] ▼	0

	Category			
A	Study	[10 B Price] := Pesavento.Swing 2 value	☐	⌐
B	Study	[11 A Price] := Pesavento.Swing 3 value	☐	⌐
C	Study	[12 B Index] := Pesavento.Swing 2 index	☐	
D	Study	[13 A Index] := Pesavento.Swing 3 index	☐	
E	Expression	[14 Square B Index] := [12 B Index] + ([12 B Index] - [13 A Index])	☐	
F			☐	
G	Study	[15 Handle] := Gann Square.Handle	☐	
H	Action	object([15 Handle]).Set Point A([12 B Index], [11 A Price])	☐	
I	Action	object([15 Handle]).Set Point B([14 Square B Index], [10 B Price])	☐	

A-B – Find the prices for the prior Pesavento swing.

C-D – Read the bar indexes for the prior Pesavento swing.

E – Calculate the index for locating the right edge of the Gann Square based on the width of the prior swing.

G-I – These rows find the Gann Square and set its two construction points.

Net Price Marker

This example displays a price marker in the right side margin that points to the last trade price, shows the daily net, and changes colors based on the current bar's close being above or below its open.

	Category	Variable	Selection #1 & #3			Show	Marker	Color	
A	Action	if (True) then marker in column 6.5				✓	☐	⬛	Last
B	Expression	[0] := Close > Open				✓		🟩	Paint Color
C	Expression	[0] := Close = Open				✓		🟨	Paint Color
D	Expression	[0] := Close < Open				✓		🟥	Paint Color
E	Action	if (True) then marker in column 3				✓	◁	⬛	Last	[^N]
F	Action	if (True) then marker in column 10				✓	▷	⬛	Last

A – Display the Rectangle marker vertically at the Last price position and filled with the Black color. The marker's horizontal location is column 6.7, which is 54 pixels to the right of the last bar's plot position.   Each column is 8 pixels in width.

B – If the Close is greater than the Open, the paint color is set to Green.

C – If the Close is equal to the Open, the paint color is set to Yellow.

D – If the Close is less than the Open, the paint color is set to Red.

E – Display the Left Triangle marker in column 3.1, using the paint color. The paint color is used because the Color selection is Black. The paint color was set to Green, Yellow, or Red by rows B, C or D.   This row also shows a label horizontally in Label Column 3. The label text is a reference tag for the daily Net for this chart's symbol. The label text also uses the paint color.

F – Display the Right Triangle marker in column 10.5, using the paint color. The vertical placement of the marker is also at the Last price, which is the vertical location used by row A and E.

Marker Size

For the 'Action | if ## then marker in column #2' selection only, the Offset entry controls the size of 32 marker selections that are icons.   An Offset of 0 selects the 16 pixel icon size.   An Offset of 1 selects a 24 pixel icon.   An Offset of 2 selects a 32 pixel icon.   Rows A, E and F in the example use Offset = 2 to show large 32 pixel icons.
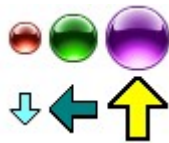
Paint Color

The paint color is set when the Show box is checked but no Marker is selected. The paint color is then used by a later row when its Color selection is Black.

Opening Gap

The Opening Gap template will show a green or red zone thoughout the day when there is a gap up or down in the opening price for the session.   The template looks for a gap at the day session open.
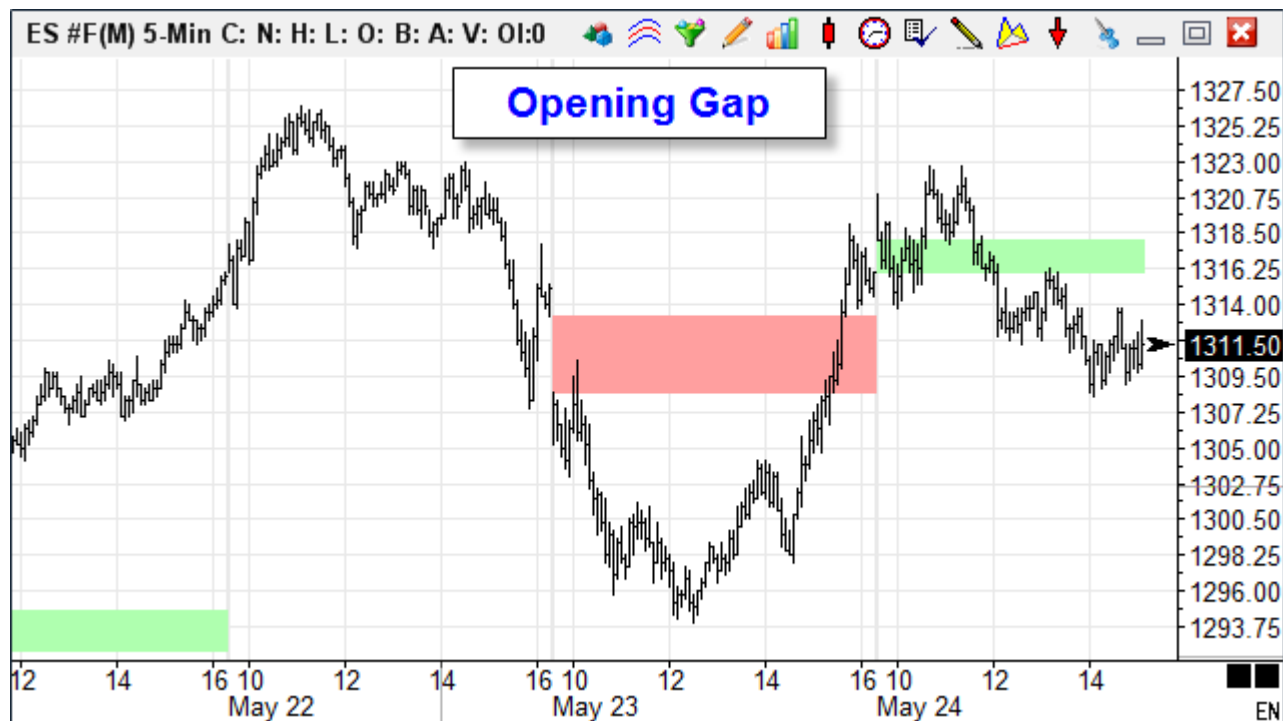
Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
Action	0	= V := #2; if ## then plot V		[Top]	0	✓	●	
		([Gap Up]			0			

A	Flag	[FirstBar] := 1st bar of Session 1	☐				
B	Branching	if Not [FirstBar] then go ahead 7 rows	☐				
C	Expression	[Gap Up] := Low > Prior High	☐				
D	Action	if ([Gap Up]) then [Top] := Low	☐				
E	Action	if ([Gap Up]) then [Bottom] := Prior High	☐				
F	Expression	[Gap Down] := High < Prior Low	☐				
G	Action	if ([Gap Down]) then [Top] := Prior Low	☐				
H	Action	if ([Gap Down]) then [Bottom] := High	☐				
I	Action	[0] := [Top]; if ([Gap Up]) then plot V	☑	●		Study Value
J	Expression	[0] := [Bottom]	☐				
K	Action	[0] := [Top]; if ([Gap Down]) then plot V	☑	●		Study Value
L	Expression	[0] := [Bottom]	☐				

A – This flag is True for the 1st bar of the Day Session.

B – For the 1st bar of the session, execute Rows C through H. Otherwise skip to Row I.

C, D, E – Set a Gap Up flag, and when True, remember the gap's 2 prices in variables named

[Top] and [Bottom].

F, G, H – Set a Gap Down flag, and when True, remember the gap's 2 prices.

I, J – When a Gap Up exists, plot a fill marker in green between the Row I and Row J prices.

K, L – When a Gap Down exists, plot a fill marker in red between the prices on Row K and L.

The Gap Up and Gap Down flags are set once per day on the 1st bar of the session.   They remain set throughout the day.   Likewise, the gap's prices are set once per day and remain unchanged throughout the day so they can be plotted by Rows I through L.

Because these flags and prices need to remain unchanged, private variables with indexes above 200 are used.   Thus, the template can be applied to other charts and not overwrite stored values.

Opening Price Principle

The Opening Price Principle is based on a statistic that the opening price is within 25% of the high or low of the daily range 65% of the time. Therefore, use this statistic to your advantage by trading in the direction of the market in relation to the opening price. When the market is trading below the open, a short position will be taken. When the market is trading above the open, a long position will be taken. This trading method can to be used with active NASDAQ and NYSE stocks trading above $20.

The Opening Price Principle can be summarized with the following rules:

1. Determine the trading range for the first hour after the day session open.
2. After one hour of trading, if the current price is below the open, look to sell short, or if the current price is above the open, look to buy long.
3. The Buy Long entry price is 0.618 of the 1st hour range down from the 1st hour high.
4. The Sell Short entry price is 0.618 of the 1st hour range up from the 1st hour low.
5. If filled, hold the trade to the end of the day and then exit. Trades are not kept overnight.
6. A protective stop is placed at one half of yesterday's range offset from the entry price.
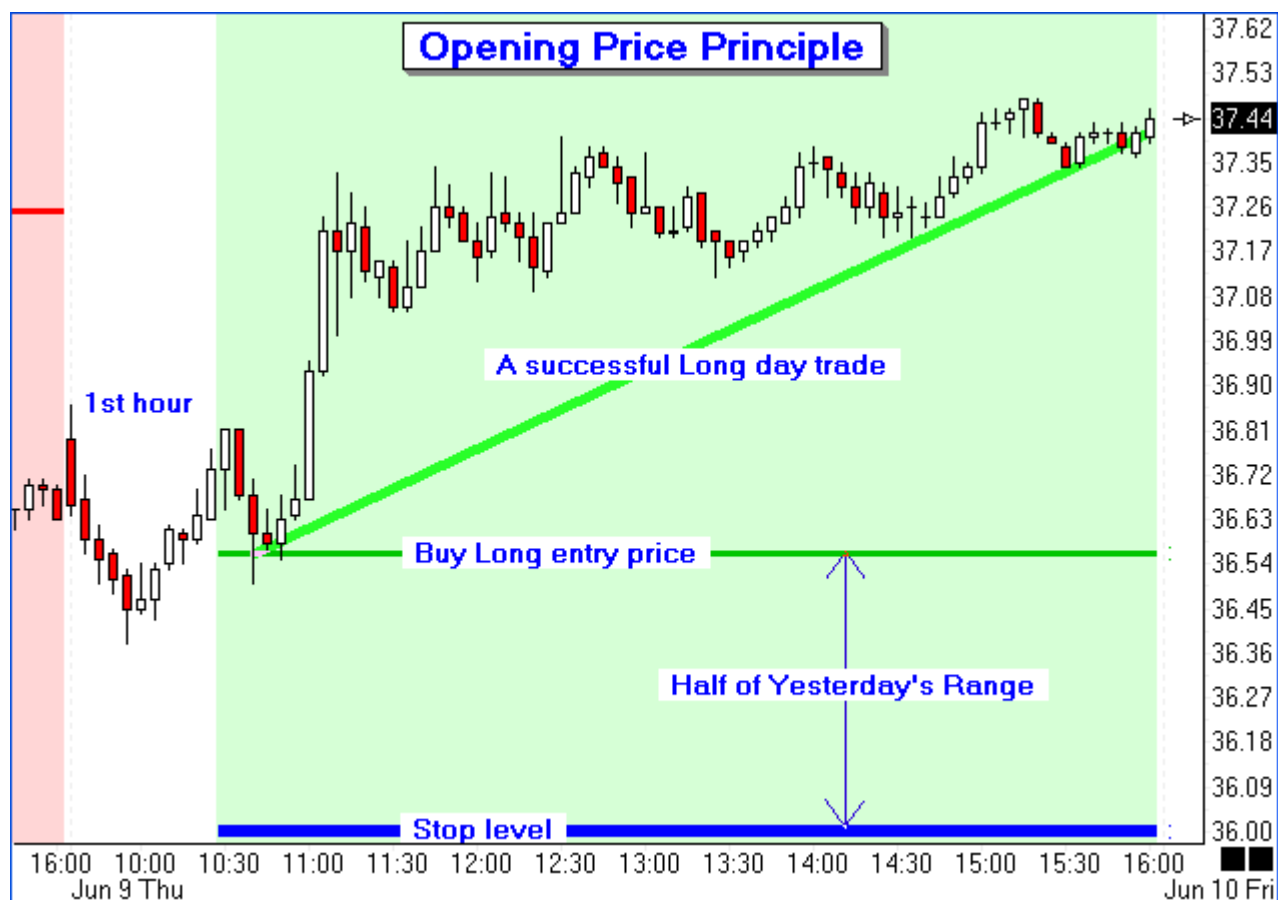
The example was hand picked to be an excellent example of the Opening Price Principle. It does not happen so perfectly every day. Often the market does not retrace to the entry price level and your Limit Order is never filled, as in the following example chart.

Opening Price Principle

Stop level

Sell Short entry price, but never filled

There will be days when the protective stop is touched and the trade is stopped out with a loss, as shown in the next example.   One should consider moving the Stop to break-even after a sufficient profit has been achieved.

Example of a Long Position that is stopped out for a 25 cent loss

However, in reality, the entry price would have been lower than the green line. And being 40 cents ahead, one should have moved the stop to break even, in my opinion.

On other days, the exit at the end of the day will be with either a small gain or a small loss, as illustrated in the next chart.
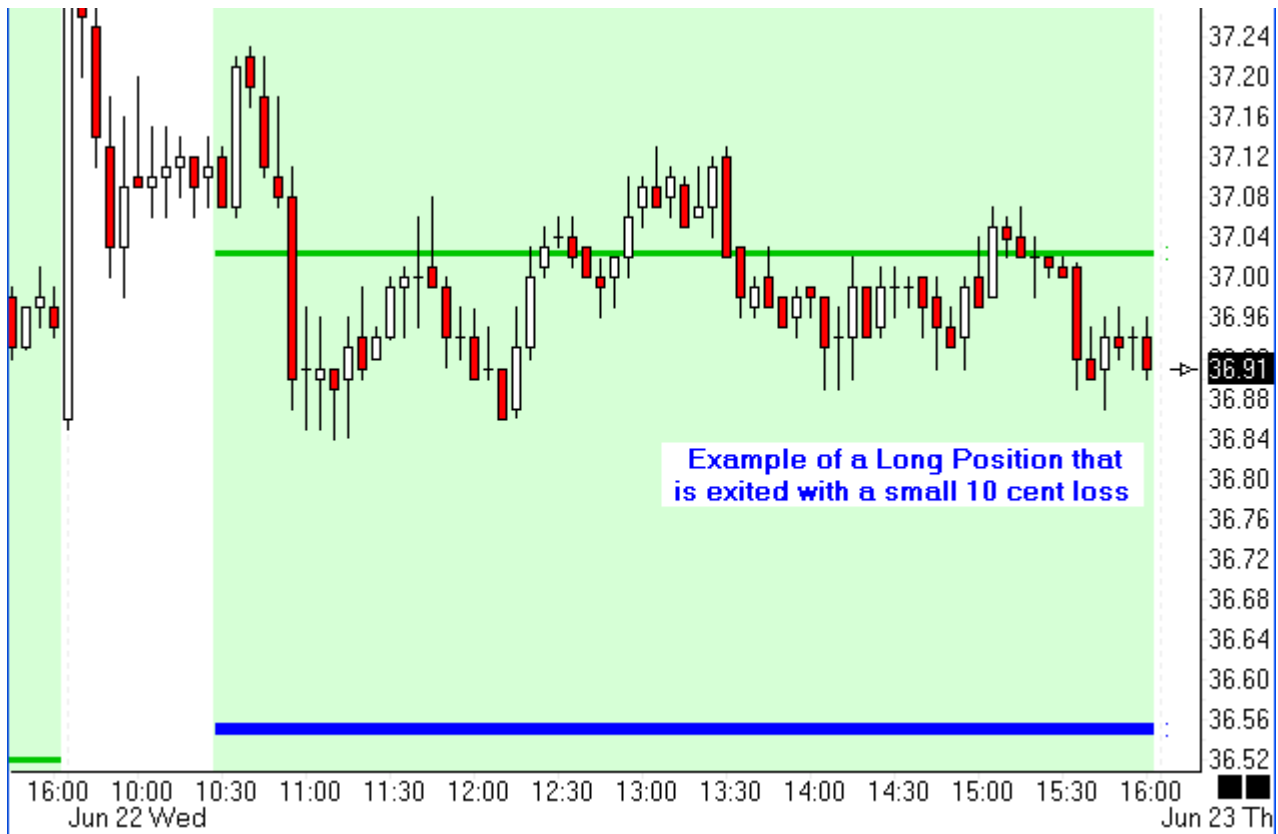
The Opening Price Principle has merit and is worth your consideration and further investigation.

read more >> DYO: Opening Price Principle

Pesavento Break

The Pesavento Patterns study is an excellent way to find and mark the swings on a chart. This example will highlight in blue the bars which break above a prior swing high in an up trend. In a down trend, the bars that trade below a prior swing low will be highlighted in yellow.

	Category	Selection #1 & #3		Show	Marker	Color
A	Study	[Swing] := Pesavento.Swing 3 value	☑	—		Study Value
B	Study	[Rising] := Pesavento.Trend line rising	☐			
C	Expression	[0] := (High > [Swing]) AND [Rising]	☑	HLT		Study Value
D	Expression	[0] := (Low < [Swing]) AND Not [Rising]	☑	HLT		Study Value

A – The prior swing price is stored in variable [Swing]. This price is marked with a horizontal pink line and represents the trigger level for the break out test.

B – The flag is True when the current Pesavento trend is rising. This result is saved in variable [Rising] for use by rows C and D.

C – Highlight a bar in blue when the bar's High is above the [Swing] price and the trend is rising. The HLT marker will paint a colored rectangle behind the bar.

D – Highlight a bar in yellow when the bar's Low is below the [Swing] price and the trend is falling.

Price and Time Bands

This example demonstrates coloring a price range and time periods.

A-B – The high and low from the 1ˢᵗ 30 minute period of the 2ⁿᵈ session is returned.   The bow tie fill marker on row A colors a band between 2 prices. The marker uses 2 prices which are obtained from the row the marker is on and the next adjacent row.

D – This illustrates marking a single time stamp with the ZON zone marker.

E – This is an example of marking a period of time before or after a session boundary.

F – This colors a period of time.   Row F is True when selection #2 (Time) is between selection #3 (1900) and selection #4 (1930).   The two times for the test are integers in the 24-hour format of hhmm.

Show Marker

The Show box must to be checked for a row's Marker and/or Label to be shown.

For rows which return prices or study values, the Marker is always shown, as in this example's row A.

For rows which return Boolean flags, the row marker is displayed only when the flag is True, as in this example's row F.

Rainbow Oscillator

The Rainbow Oscillator is similar to the Stochastic formula, where the current bar's close is evaluated in a maximum range for a period of bars. The difference is the close is measured from an average instead of from the bottom of the range.

Formula

Rainbow = 100 * (Close – Average) / (Highest Close – Lowest Close)

Stochastic = 100 * (Close – Range Low) / (Range High – Range Low)

The Average is calculated by nesting averages of averages, and then averaging these.

	Study Name	Study Location	Marker Location	Label Location	Grid Tab
	Rainbow Oscillator	Window 1	Study Value	None	Default

	Study Scale			Variables File	Message Location
	Data 0 Centerline			-Rainbow OSC	None

Message Text

Font   True   False   Panel

Sound
- ● Silent
- ○ Beep
- ○ Voice
- ○ WAV
- ☐ Sound once per bar

WAV file

Browse

Trading System

Price: Close

Quantity: 1

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Expression	14 Rainbow	= Close	-	[13 Average]	0
		/ ([11 Max Close]	-	[12 Min Close]	0)

	Category	Expression/Function
A	Expression	[201 Period] := 6
B	Function	[1] := Simple Average(Close, [201 Period])
C	Function	[2] := Simple Average([B], [201 Period])
D	Function	[3] := Simple Average([C], [201 Period])
E	Function	[4] := Simple Average([D], [201 Period])
F	Function	[5] := Simple Average([E], [201 Period])
G	Function	[6] := Simple Average([F], [201 Period])
H	Function	[13 Average] := Simple Average([1], [201 Period])
I	Function	[11 Max Close] := Highest(Close, [201 Period])
J	Function	[12 Min Close] := Lowest(Close, [201 Period])
K	Expression	[14 Rainbow] := (Close - [13 Average]) / ([11 Max Close] - [12 Min Close])
L	Expression	[0] := 100 * [14 Rainbow]

A – Enter a value for the Period to use for the averages and number of bars in the range.

B – Calculate a simple average of a data point such as the Close. Average uses Period.

C – Calculate an average of the average on Line B.

D – Calculate an average of the average on Line C.

E – Calculate an average of the average on Line D.

F – Calculate an average of the average on Line E.

G – Calculate an average of the average on Line F.

H – 6 averages were calculated on Line B – G, and their values saved in Variables [1] – [6]. This line now averages these 6 averages by referencing the starting variable [1].

I – Find the Maximum Close for the prior Period number of bars.

J – Find the Minimum Close for the prior Period number of bars.

K - L – Calculate and plot the Rainbow Oscillator formula

Range Completion

This example shows a band on the chart where the Range bar's close would have to progress to to complete the Range bar.   A price outside of the band would start a new Range bar.
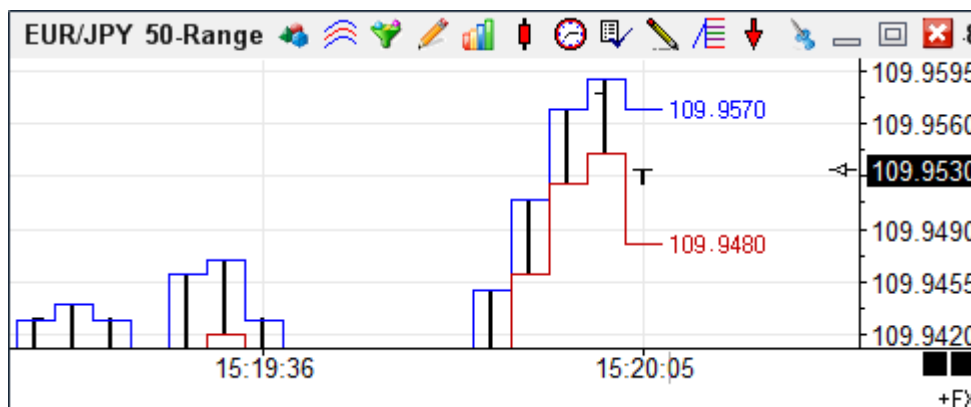
In this example for a 50 Range bar, the bar under construction would have to move up to 109.9570 or down to 109.9480 to complete the bar's range and start a new bar. These values will adjust as the bar develops.   A lower low will move down the top band level, and a higher high will move up the bottom band level.

A – Abort the calculation and plotting except when working with the last bar on the chart.

B – The top band is calculated by adding the range bar size to the bar's low, and plotted.

C – The bottom band is calculated by subtracting the range size from the bar's high.

Relative Slope

Formula

$K = EMA((H+L+C)/3,10)$
$S1 = 2*(K-REF(K,-1))/(K+REF(K,-1))$
$RS = 100*EMA(S1,3)$

Category	Variable	Selection #1 & #3		Op. [#]		Selection #2 & #4		Offset	
Expression ▾	0	▾	= [20 Average]	▾	- ▾	[A]	▾	-1 ▲▼	
크 弖 📁	/	▾	([20 Average]	▾	+ ▾	[A]	▾	-1 ▲▼	)

A	Function	[20 Average] := Exponential Ave((H+L+C)/3, 10)		☐	
B	Expression	[0] := ([20 Average] - [A][-1]) / ([20 Average] + [A][-1])		☐	
C	Function	[20 Average] := Exponential Ave([B], 3)		☐	
D	Expression	[0] := [20 Average] * 200		☑ ⌒ 🟫	Study Value

A – Calculate a 10 period Exponential average of the (H+L+C)/3

B – Implements the S1 formula. [A] is the value from Row A. [A][-1] is the prior value.

C – Calculate an Exponential average of S1 which is the Row B values.

D – Calculate the final RS to plot. The multiplier is 200 because 2 was not done on S1.

Scoreboard

This example reads bar and study values from 5 charts and displays these values on a spreadsheet.

	ES #F	B	C	D	E	F	G	H	I	J
1	1173.00	High	Low	prior High	prior Low	STO %K	%K/%D	RSI	RSI/AVE	TREND
2	5 min bar	1173.50	1173.00	1173.50	1172.75	93.65	Below	61.85	Above	↑
3	60 min bar	1173.50	1169.75	1171.50	1168.25	27.41	Above	43.04	Above	↓
4	Daily	1186.25	1167.75	1193.00	1180.50	71.35	Below	56.38	Below	↑
5	Weekly	1193.00	1167.75	1186.25	1155.50	93.10	Below	65.91	Below	↑
6	Monthly	1193.00	1127.00	1153.50	1050.75	54.48	Above	59.39	Above	↑

Here is the property form for the spreadsheet.

The next image shows the cell formula used.

	ES #F	B	C	D	E	F	G	H	I	J
		High	Low	prior High	prior Low	STO %K	%K/%D	RSI	RSI/AVE	TREND
1	=Last("ES									
2	5 min bar	=Get(100)	=Get(101)	=Get(102)	=Get(103)	=Get(104)	=IF(Get(1	=Get(106	=IF(Get(	↑
3	60 min bar	=Get(110)	=Get(111)	=Get(112)	=Get(113)	=Get(114)	=IF(Get(1	=Get(116	=IF(Get(	↓
4	Daily	=Get(120)	=Get(121)	=Get(122)	=Get(123)	=Get(124)	=IF(Get(1	=Get(126	=IF(Get(	↑
5	Weekly	=Get(130)	=Get(131)	=Get(132)	=Get(133)	=Get(134)	=IF(Get(1	=Get(136	=IF(Get(	↑
6	Monthly	=Get(140)	=Get(141)	=Get(142)	=Get(143)	=Get(144)	=IF(Get(1	=Get(146	=IF(Get(	↑

Most cells are showing Variable values using the =Get(index) function. The variables are being set by a DYO present on each of the 5 charts. An example DYO will be shown later.

Cell A1 has the function: =Last("ES #F","ES")

The rest of the cells on row 1 and in column A contain text as shown.

Column G and I use the =IF function to display text based on a Boolean value. The formula in those 2 columns are shown next. Note how a function is used in the 1st parameter position.

G
%K/%D
=IF(Get(105),"Above","Below")
=IF(Get(115),"Above","Below")
=IF(Get(125),"Above","Below")
=IF(Get(135),"Above","Below")
=IF(Get(145),"Above","Below")

I
RSI/AVE
=IF(Get(107),"Above","Below")
=IF(Get(117),"Above","Below")
=IF(Get(127),"Above","Below")
=IF(Get(137),"Above","Below")
=IF(Get(147),"Above","Below")

Each chart contains a Stochastic, a Relative Strength, an Auto Trends study, and a DYO. The DYO on the 5-minute chart uses variables [100] through [108] to populate row 2 in the spreadsheet. The other charts have similar DYOs, where the only thing that differs are the variables being assigned. The 60-minute chart uses variables [110] through [118], etc.

A	Expression	[100] := High
B	Expression	[101] := Low
C	Expression	[102] := Prior High
D	Expression	[103] := Prior Low
E	Study	[104] := STO.%K value
F	Study	[105] := STO.%K >= %D
G	Study	[106] := RSI.RSI value
H	Study	[107] := RSI.RSI >= Average
I	Study	[108] := Trend.Swing direction
J	Action	if [108] then Marker(10, 2) := Marker
K	Action	if [108] = 0 then Marker(10, 2) := Marker

Row F and H are storing Boolean flags for use in the spreadsheet's =IF functions.

Row I is storing a Boolean flag used by row J and K to place the arrow markers in Column J. The row K test of 'if [108] = 0 then' is testing for the flag to be False.

Spreadsheet Statements

The DYO Action category has the following statements for use with a spreadsheet.

Find Spreadsheet – The text in the Label field, or Message field if the Label is blank, is the name of the spreadsheet to find. If the spreadsheet is not found, this statement will open a spreadsheet form with the name.

Cell(col, row) := Format(#2, [#]) - Use this statement to place a value from Selection #2 into a cell. The value is formatted with [#] decimals, or as a TDateTime.

```
Find Spreadsheet( Message )
Cell( #3, #4 ) := Format( #2, [#])
Fontsize( #3, #4 ) := #2
Fontbold( #3, #4 ) := #2 oper 0
if #2 oper 0 then Fontcolor( #3, #4 ) := C
if #2 oper 0 then Cell( #3, #4 ) := Messa

if ## then Rowcolor( #2 ) := Color
Label := Cell( #3, #4)
V := Cell( #3, #4 ) oper #2
V := Cell( #3, #4 ) row move #2 cells
V := Cell( #3, #4 ) col move #2 cells
Cell( #3, #4 ) := Format( V, [#]) row of #2
Cell( #3, #4 ) := Format( V, [#]) col of #2
Spreadsheet.Recalculate
Spreadsheet.Clear
Spreadsheet.Size( #3 cols x #4 rows)
```

[#]	Format
0	Integer
1 to 6	Number of Decimals
7	TDateTime hh:nn
8	TDateTime hh:nn:ss
9	TDateTime mm-dd-yy
10	Grid's default decimals

Fontsize(col, row) - Changes the font size for the cell.   Size is the Selection #2 value.

Fontbold(col, row) - Changes the font style for the cell.   Bold when the expression is True.

Fontcolor(col, row) - If the #2 oper 0 expression is True, the cell font color will be set to the DYO row's color selection. This is an override to the spreadsheet's font color.

Cell(col, row) := Message – If the #2 oper 0 expression is True, the text in the Label field, or Message field if the Label is blank, is placed in the cell.   This text could be a formula or function. The text can contain reference tags.

Also, the statement can place a Marker in a cell.   There is no control over the marker color. Markers will show as seen on the Marker drop down list, where most markers are black.

The cell background color is set to the DYO row's color selection when the Show box is checked. This is an override to the spreadsheet's color or band coloring.

The following line writes the chart's symbol in cell(1, 0), which is the 'ES #F' in the Scoreboard example. [$S] is the reference tag for the chart's symbol.

Spreadsheet	SCOREBOARD	
ES #F	B	C
1 1174.50	High	Low

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
Action	0	= if #2 oper 0 then Cell(#3, #4) :		True	0			
ESPL		(1		0	0	)	[$S]	

Rowcolor(row) - If the ## expression is True, the row's background color is set to the DYO row's color selection. This is an override to the spreadsheet's color or band coloring.

Label := Cell(col, row) - A cell's content is assigned to a DYO Label. The DYO label text can be displayed in the margin of the chart.

V := #2 oper Cell(col, row) - A cell's value is used in the expression with Selection #2 and assigned to the variable V.

V := Cell(col, row) row move #2 cells – This statement copies a series of cell values from a spreadsheet row beginning with Cell(col,row) into the Variables array beginning with V. #2 is the number of cells to copy.

V := Cell(col, row) col move #2 cells – This statement copies a series of cell values from a spreadsheet column beginning with Cell(col,row) into the Variables array beginning with V. #2 is the number of cells to copy.

Cell(col, row) := Format(V, [#]) row of #2 cells - Use this statement to place a series of values from the Variables array, beginning with V into a row of cells beginning with Cell(col,row). Selection #2 is the number of cells to fill. The values are formatted with [#] decimals, or as a TDateTime.

Cell(col, row) := Format(V, [#]) col of #2 cells - Use this statement to place a series of values from the Variables array, beginning with V into a column of cells beginning with Cell(col,row). Selection #2 is the number of cells to fill. The values are formatted with [#] decimals, or as a TDateTime.

Spreadsheet.Recalculate – The spreadsheet update mode could be on Manual, and the DYO controls when the spreadsheet calculates. Perhaps the DYO is updating Variables or cells on the spreadsheet, then needs the spreadsheet to calculate before spreadsheet results are read by the DYO and used in DYO expressions.

Spreadsheet.Clear – This statement will clear (erase) the contents of the spreadsheet.

Spreadsheet.Size – This statement can be used to change the number of columns and rows.

Stochastic Trade System

This example is a trading system based on the Stochastic study using the Design Your Own (DYO) feature in Ensign 10.   See Ensign's Risk Disclosure and Disclaimer.

Let's begin by understanding the fundamental behavior of Stochastic so that we can pick some rules for when to enter and exit a position.   Click this link for additional reading on the Stochastic Fundamental Behavior.
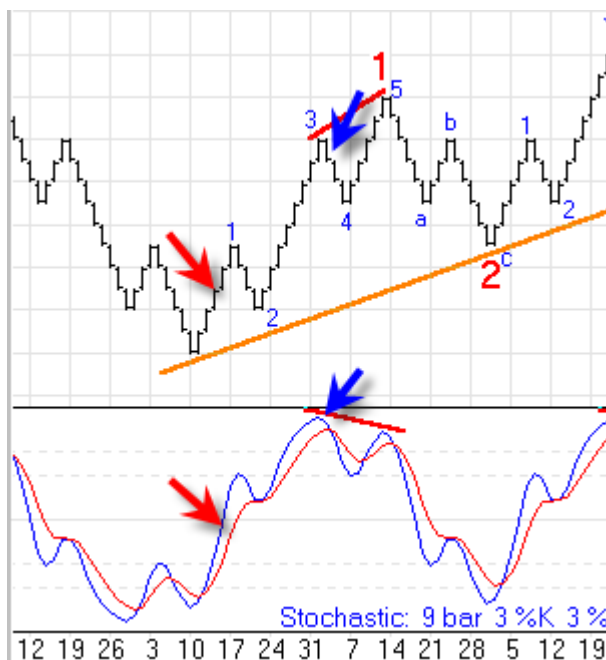
I want to catch a new trend when it starts, which has been marked with the Red arrows on the chart.   The Buy Signal could be based on the Stochastic %K line crossing the 50% level after having been below 30%.   One of the exit rules or objectives could be when the %K line turns down the 2nd time, which is marked by the Blue arrows. This would typically be at the end of a 3rd wave in the Elliott wave pattern.   So the portion of the chart I am trying to scalp is from the Red arrow entry to the Blue arrow exit.   The inverted pattern would be the basis for the Sell Signal.

I would also like the system to try to scalp a trade based on a profit objective. So I will design the system to do a blend of both, and have rules for aborting the trade based on either an adverse move or running out of time at the end of the day session.   Hopefully in a choppy market the 1st contract's profit objective will put some money in the bank to offset small losing trades, and in a trending market the 2nd contract held for the 2nd Stochastic turn will make a nice profit for the system.

The following is an outline of the rules for the Stochastic Trade System.   The design is based on trading the EUR/USD forex symbol, but could be adapted to trading any symbol, including the popular ES, ER2, NQ or YM futures symbols.

General Rules

1. Trade signals between 7 a.m. and 3 p.m. Eastern Zone time.   That is when I am watching the markets.
2. Exit entire position (if any) at 4 p.m.   I do not want to carry positions over night.
3. Exit 1 contract on a profit objective of 10 pips.   Test the system with larger and smaller profit objectives.

Long Trade Rules

1. The Buy Signal will be %K being below 30 and then crossing above 50.   Buy 2 contracts.
2. Exit entire position which may be 1 or 2 contracts on the 2nd %K downturn. Location of the downturn does not matter.
3. Exit entire Long position if %K crosses below 40.   This is the bail out rule for being on the wrong side of the move.

Short Trade Rules

1. The Sell Signal will be %K being above 70 and then crossing below 50.   Sell 2 contracts.
2. Exit entire position which may be 1 or 2 contracts on the 2nd %K upturn. Location of the upturn does not matter.
3. Exit entire Short position if %K crosses above 60.   This is the bail out rule for being on the wrong side of the move.

Time Conditions
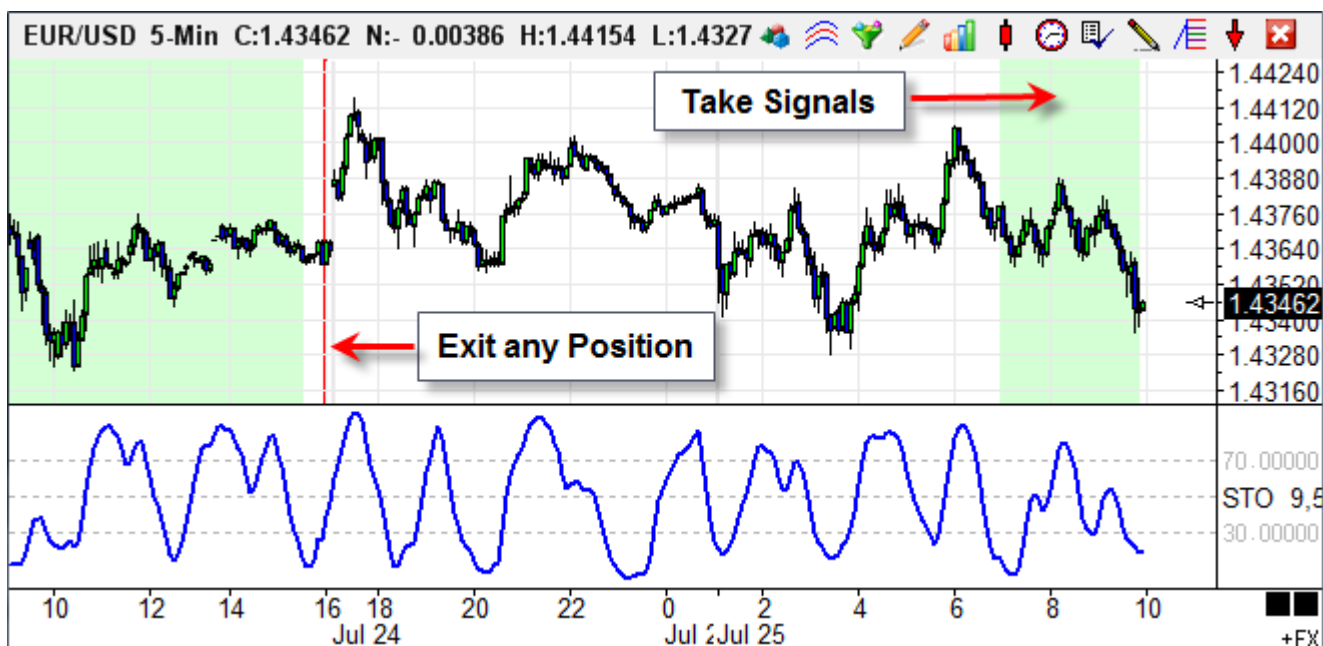
Let's begin by taking each rule, one at a time, and show its implementation using DYOs in Ensign 10.

This DYO implements Boolean flags for the two rules based on Time. Signals will only be taken between 7:00 and 15:00 EST which is marked on the chart with the green background. This time condition will be stored as a flag in Variable [10] and named Time Rule.

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
Expression ▼	10 Time Rule ▼	= Time	▼ >= ▼	700	▼ 0 ▲	✓	ZON ▼	
▤ ▤ ▤	AND ▼	(Time	▼ <= ▼	1500	▼ 0 ▲)|			

A	Expression	[10 Time Rule] := (Time >= 700) AND (Time <= 1500)	✓ ZON		Study Value
B	Action	if (Time = 1600) then Exit Position	✓ ZON		Study Value

Buy Signal
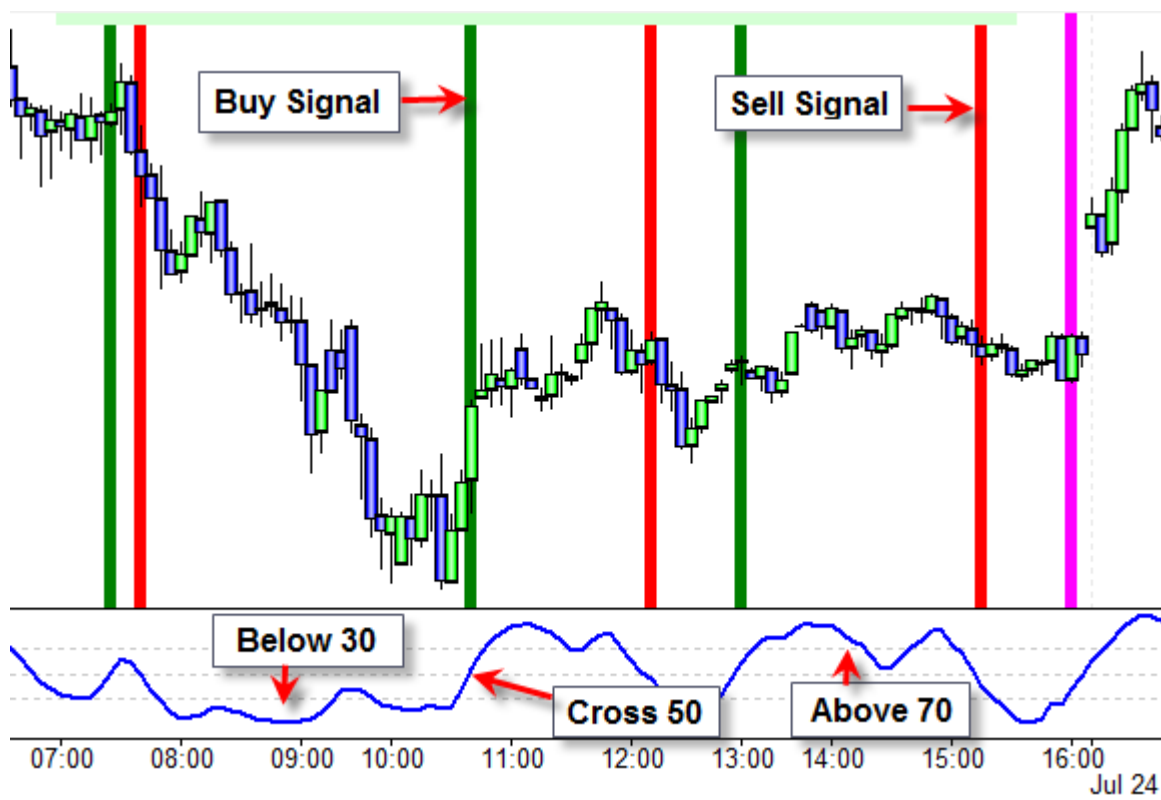
C – Variable 11 is True when Stochastic %K is below 30.

D – Variable 12 is True when Stochastic %K crosses above 50.

E – When %K is below 30, a Flag in [250] is set to true to remember the event. This private variable is used to remember this event between calls since the trade system might be running on multiple charts simultaneously. The flag in [11] will not be True when [12] is True.

F – This is the Buy signal based on the Time Rule, and %K has been below 30, and %K is now crossing above 50.

G – When there is a Buy signal on Row F, reset the flag in [250]. The condition for K% below 30 will have to set itself again before another Buy signal is permitted.

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
Action	252 Buy Signal ▼ =	if # then Buy or Reverse ▼	▼	[10 Time Rule] ▼	0	✓	Z0N ▼	
	AND ▼	([250 StoBuySet] ▼	AND ▼	[12 StoX>50] ▼	0)			

A	Expression	[10 Time Rule] := (Time >= 700) AND (Time <= 1530)	✓	▬		Top Row 1	
B	Action	if (Time = 1600) then Exit Position	✓	Z0N		Study Value	
C	Study	[11 Sto<=30] := (STO.%K value) <= 30	☐	Z0N			
D	Study	[12 StoX>50] := (STO.%K value) X> 50	☐	Z0N			
E	Action	if ([11 Sto<=30]) then [250 StoBuySet] := True	☐	Z0N			set
F	Action	if [10 Time Rule] AND ([250 StoBuySet] AND [12 StoX>50]) then Buy or	✓	Z0N		Study Value	
G	Action	if ([252 BuySignal]) then [250 StoBuySet] := False	☐				reset
H	Study	[13 Sto>70] := (STO.%K value) >= 70	☐	Z0N			
I	Study	[14 StoX<50] := (STO.%K value) X< 50	☐	Z0N			
J	Action	if ([13 Sto>70]) then [251 StoSellSet] := True	☐	Z0N			set
K	Action	if [10 Time Rule] AND ([251 StoSellSet] AND [14 StoX<50]) then Sell or	✓	Z0N		Study Value	
L	Action	if ([253 Sell Signal]) then [251 StoSellSet] := False	☐				reset

Sell Signal

The Sell signal is the inverse logic of the Buy signal. %K must have been above 70 and then cross below 50.

H – Variable 13 is True when Stochastic %K is above 70.

I – Variable 14 is True when Stochastic %K crosses below 50.

J – When %K is above 70, a Flag in [251] is set to true to remember the event. This private variable is used to remember this event between calls since the trade system might be running on multiple charts simultaneously. The flag in [13] will not be True when [14] is True.

K – This is the Sell signal based on the Time Rule, and %K has been above 70, and %K is now crossing below 50.

L – When there is a Sell signal on Row K, reset the flag in [251]. The condition for K% above 70 will have to set itself again before another Sell signal is permitted.

Profit Exit

The next DYO tests for a Profit scalp objective being met.   The DYO will only mark conditions where the Position Size is either a 2 for a Long or -2 for a Short.   This means that the scalp to take off the 1st contract has not happened yet.   The scalp size in points is set on Row A. The DYO show a thin green stripe on the chart where the Position Size is 2 and the profit exceeds 0.0010 points. This is where the long position is reduced by selling 1 contract.

The thin red stripes show where the position is short 2 contracts (position size = -2) and the profit objective of 0.0010 points has been met. This is where the short position is reduced by buying 1 contract.
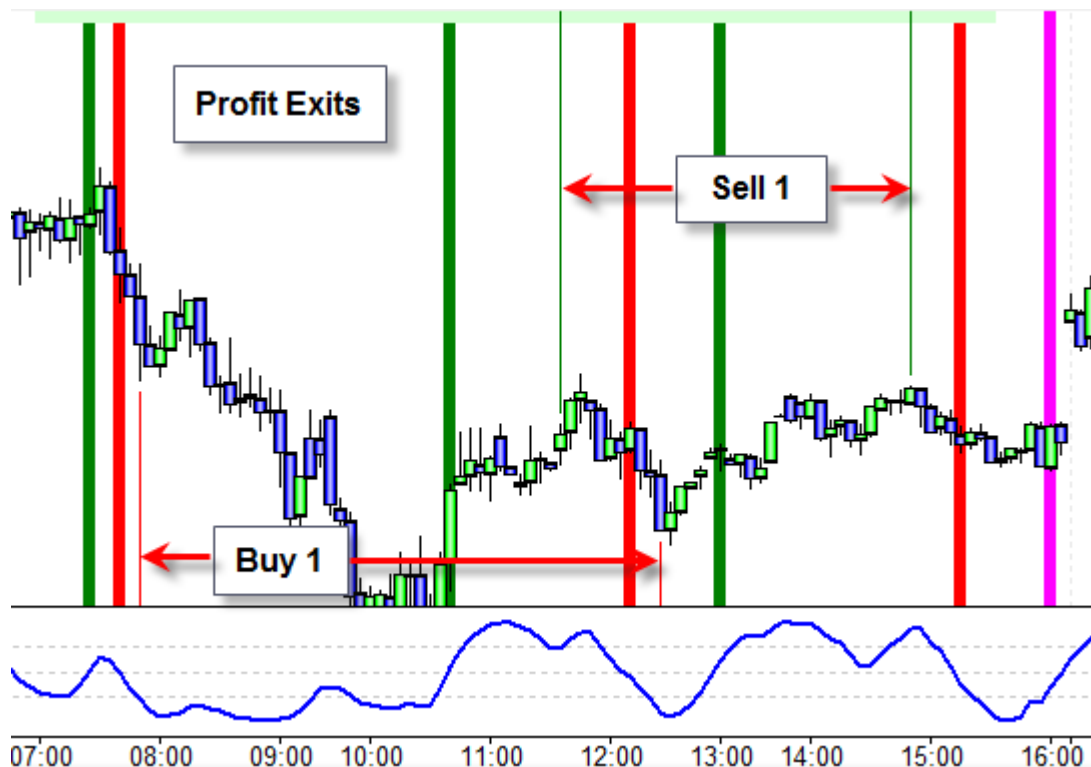
Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
Action	0	= if # then Sell Market		[16 Profit Flag]	0	✓		
	AND	(Position Size	=	2	0			

A	Action	if (Not Optimizer Running) then [15 Points Objective] := 0.001	☐		
B	Expression	[16 Profit Flag] := Profit Points >= [15 Points Objective]	☐	—	
C	Action	if [16 Profit Flag] AND (Position Size = 2) then Sell Market	✓		Above High
D	Action	if [16 Profit Flag] AND (Position Size = -2) then Buy Market	✓		Below Low

A – Set the Points objective.   The statement to conditionally set the points objective is used in anticipation of using the program's Optimizer feature to optimize this parameter. For this example, a fixed value of 0.001 points for the EUR/USD chart is used.

B – Variable [16] is True when the Profit objective has been achieved.

C – Reduce the Long position by selling 1 contract at the market.

D – Reduce the Short position by buying 1 contract at the market.

Stochastic Exit

The Stochastic Exit watches for the 2nd downturn of a rising %K line and uses that signal to exit any Long position.   The 2nd upturn of a falling %K line is a signal to exit any Short position.
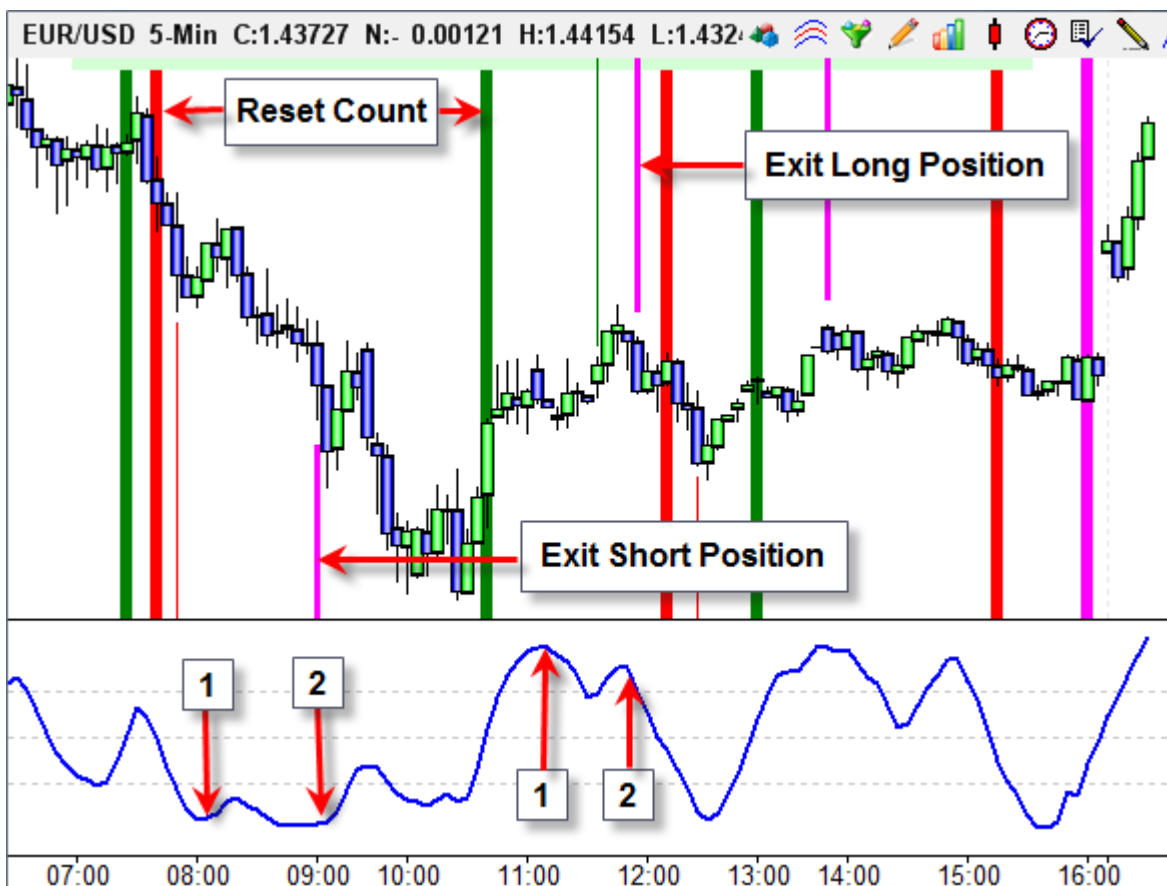
Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
Action ▾ 0	▾	= if # then Exit Long Position ▾	▾	Position is Long	▾	0 ⏶	☑	⊓ ▾	
	AND ▾	([254 Top Count]	▾ >= ▾	2	▾	0 ⏶)			

A	Action	if ([252 Buy Signal] OR Position is Flat) then [254 Top Count] := 0	☐				reset
B	Study	[17 Turn Down] := STO.%K turns down	☐	▼			
C	Action	if [17 Turn Down] then inc([254 Top Count])	☐				
D	Action	if Position is Long AND ([254 Top Count] >= 2) then Exit Long Position	☑	⊓		Above High	
E			☐				
F	Action	if ([253 Sell Signal] OR Position is Flat) then [255 Bottom Count] := 0	☐				reset
G	Study	[18 Turn Up] := STO.%K turns up	☐	▲			
H	Action	if [18 Turn Up] then inc([255 Bottom Count])	☐				
I	Action	if Position is Short AND ([255 Bottom Count] >= 2) then Exit Short	☑	⊓		Below Low	

A – Reset the [254] variable used to count the tops in the Stochastic curve.

B, C – Increment the [254] variable when the %K line turns down.

D – Exit any Long position when the top counter reaches 2.

71

Wrong Trend Exit

The system will exit when the Stochastic is moving counter to the Position.   Exit any Long position when the %K crosses below 40. Exit any Short position when the %K crosses above 60.

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
Action ▾	0 ▾	= if # then Exit Short Position ▾	▾	Position is Short ▾	0 ▴▾	☑	ili ▾	▉
🔀	AND ▾	([19 StoX>60] ▾	▾	▾	0 ▴▾	)		

A	Study	[19 StoX>60] := (STO.%K value) X> 60		☐			
B	Action	if Position is Short AND [19 StoX>60] then Exit Short Position		☑	ili	▉	Below Low
C				☐			
D	Study	[20 StoX<40] := (STO.%K value) X< 40		☐			
E	Action	if Position is Long AND [20 StoX<40] then Exit Long Position		☑	⌐	▉	Above High

A – Variable [19] is True when the %K crosses above 60 and the Position is Short.

B – Exit any Short position when [19] is True.

D – Variable [20] is True when the %K crosses below 40 and the Position is Long.

E – Exit any Long position when [20] is True.

Trade Report

Click the sub-menu Trade Report on the Account button on the Main ribbon.

The top portion of the form shows summary information. The Winners and Losers count needs to be taken with a 'grain of salt' because of the way the ledger logs our method of putting on 2 contracts, and then possibly taking off one contract on a profit exit.   Rows 45 and 46 show this effect.   Row 45 shows a Short Position of Size 2 with an entry price of 1.43942. The profit objective was met on a later bar that closed at 1.43814. The Notes show this trade was on from 07:40 until 07:50. The 2nd contract traded is shown on row 46. It was put on at the same price of 1.43814 at 07:50, and removed at 09:00 at a different price of 1.43617. The total for this short trade with a split exit was $256+$197 = $453.

	Symbol	Pos	Quantity	EntryDate	EntryPrice	ExitPrice	Profit/Loss	Notes
44	EUR/USD	L	2	07-22-11	1.44052	1.43942	$ -220.00	07:25 - 07:40
45	EUR/USD	S	2	07-22-11	1.43942	1.43814	$ 256.00	07:40 - 07:50
46	EUR/USD	S	1	07-22-11	1.43814	1.43617	$ 197.00	07:50 - 09:00
47	EUR/USD	L	2	07-22-11	1.43546	1.43651	$ 210.00	10:40 - 11:35
48	EUR/USD	L	1	07-22-11	1.43651	1.43603	$ -48.00	11:35 - 11:55
49	EUR/USD	S	2	07-22-11	1.43659	1.43474	$ 370.00	12:10 - 12:25
50	EUR/USD	S	1	07-22-11	1.43474	1.43624	$ -150.00	12:25 - 13:00
51	EUR/USD	L	2	07-22-11	1.43624	1.43675	$ 102.00	13:00 - 13:50
52	EUR/USD	S	2	07-22-11	1.43630	1.43665	$ -70.00	15:15 - 16:00
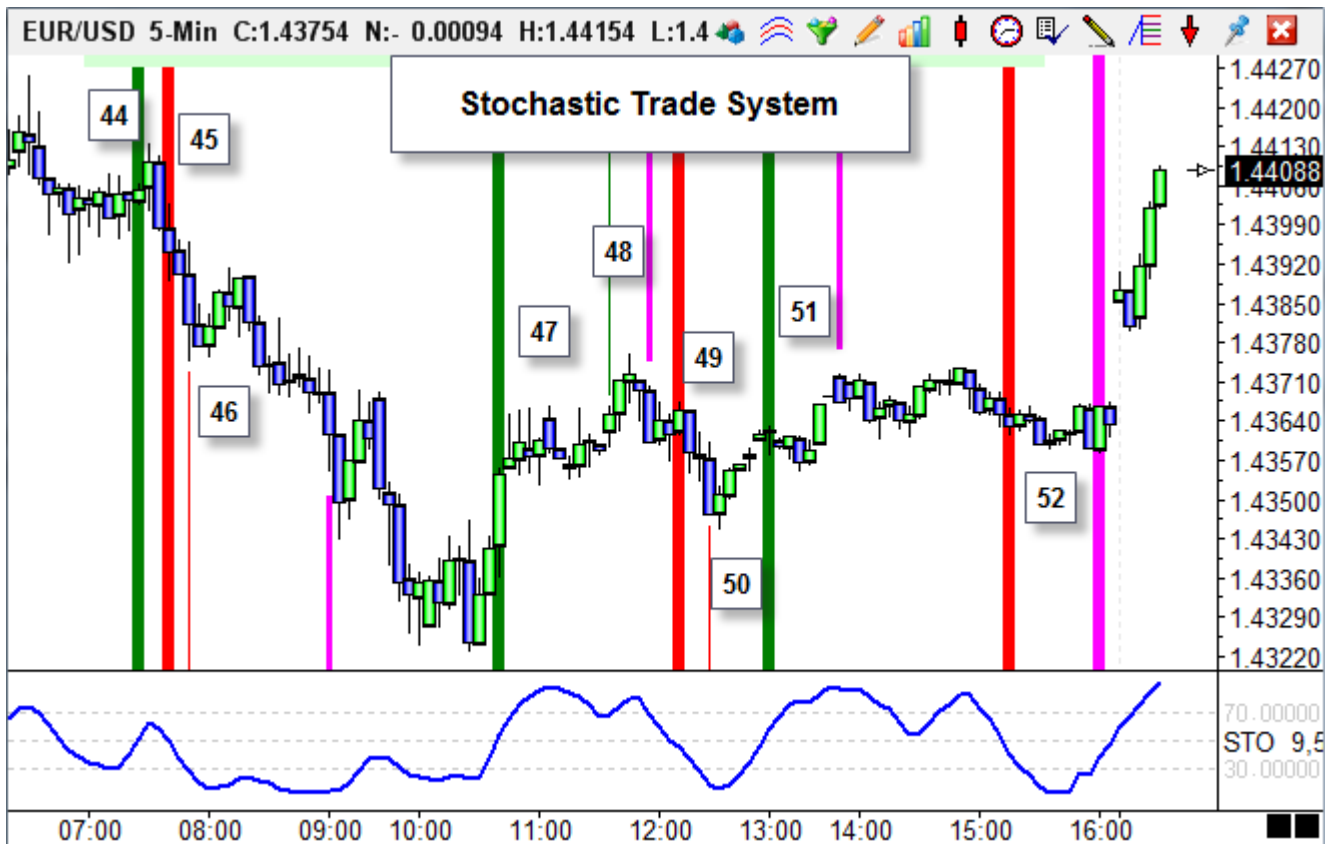
The trades from the Trade Report ledger are marked by their ledger row number on the chart.

System Optimization

One of the unknowns was what scalp objective might be the most profitable. The system was initially designed with a 0.0010 pip profit object for the EUR/USD. We will conclude this example by using the Optimize Trades feature to research the best objective to use.

The Optimizer has been configured to test one parameter, and that is variable [15] which was used in the Stochastic Trade System to hold the Points Objective. The original value was 0.0010 pips. The Optimizer will test the system with a range of values from 0.0006 to 0.0030 in steps of 0.0002. The following report was generated for a test chart.

	Chart			Scalp	Win #	Loss #	Total #	% Win	$ Win	$ Loss	Profit	Ratio	Average
1	EUR_USD.5	0.0000	0.0000	0.0006	8	11	19	42	1296	-1472	-176	0.8805	-9.2613
2	EUR_USD.5	0.0000	0.0000	0.0008	7	11	18	39	1365	-1486	-121	0.9186	-6.7201
3	EUR_USD.5	0.0000	0.0000	0.0010	7	11	18	39	1603	-1605	-2	0.9988	-0.1086
4	EUR_USD.5	0.0000	0.0000	0.0012	7	10	17	41	1801	-1704	97	1.0570	5.7087
5	EUR_USD.5	0.0000	0.0000	0.0014	7	10	17	41	1843	-1704	139	1.0816	8.1799
6	EUR_USD.5	0.0000	0.0000	0.0016	7	10	17	41	1843	-1704	139	1.0816	8.1799
7	EUR_USD.5	0.0000	0.0000	0.0018	7	9	16	44	1656	-1594	62	1.0389	3.8788
8	EUR_USD.5	0.0000	0.0000	0.0020	7	8	15	47	1356	-1444	-88	0.9391	-5.8627
9	EUR_USD.5	0.0000	0.0000	0.0022	7	8	15	47	1459	-1478	-19	0.9872	-1.2628
10	EUR_USD.5	0.0000	0.0000	0.0024	7	7	14	50	1208	-1346	-138	0.8975	-9.8535
11	EUR_USD.5	0.0000	0.0000	0.0026	6	8	14	43	1282	-1346	-64	0.9525	-4.5674
12	EUR_USD.5	0.0000	0.0000	0.0028	6	8	14	43	1282	-1346	-64	0.9525	-4.5674
13	EUR_USD.5	0.0000	0.0000	0.0030	6	8	14	43	1282	-1346	-64	0.9525	-4.5674

The best value in each column is highlighted in yellow. The Scalp parameter that generated the highest Profit was 0.0014 which is a few pips higher than the original parameter.

Summary

This example is shown to demonstrate designing a trading system in Ensign 10 using studies and the Design Your Own tool, which is very flexible, powerful and efficient. Though this system is probably unprofitable, it is of sufficient complexity to demonstrate several things that might be included in a trading system of your own design.

The following items might improve the Stochastic Trade System and would be areas worthy of further research.

- Try other parameters for the Stochastic study. A faster stochastic might enter a trade nearer the swing turn, but it might also exit trending trades too soon.
- Try other chart symbols and chart time frames.
- Possibly the Wrong Trend Exit should be based on points lost, instead of waiting for a counter move in the Stochastic %K line.

Various ideas, such as the above, lend themselves to using the Optimizer to aid in the research of what range of values are 'in the ballpark.' Best of luck in your efforts to design your own trading system.

Study Consensus

This example shows how a Buy or Sell signal can be generated when three studies agree with each other. The purpose of the example is to show how consensus among studies can be determined, and no claim is being made that the signals are profitable.

The example will use input from three studies: Relative Strength Index (RSI), Stochastic Momentum (SM), and two Moving Averages (MA).   The study signals are defined as:

- Buy when 3 period MA > 17 period MA, Sell when 3 period MA < 17 period MA.
- Buy when SM > Ave(SM), and Sell when SM < Ave(SM). Use 10 Bar, 25, 2, 7 averages.
- Buy when RSI > 60, and Sell when RSI < 40.   Use a 7 period RSI parameter.
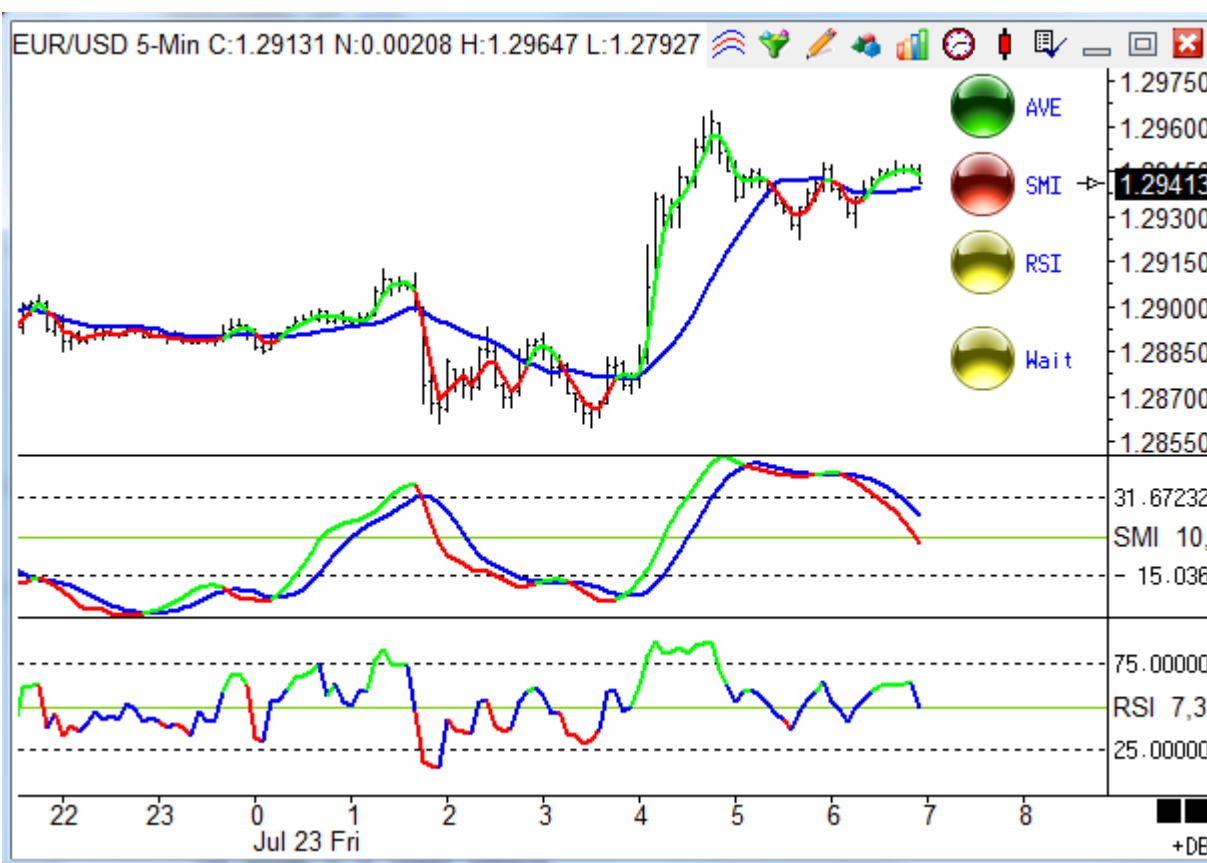
The 3 period moving average is plotted in Green on the chart when it is above the 17 period moving average plotted in Blue. The 3 period MA is plotted in Red when it is below the 17 period MA.

The first study panel shows the Stochastic Momentum study plotted in Green and Red, based upon whether the study is above or below a 7 period average of the SM plotted in Blue.

The bottom panel shows the Relative Strength Index. It is plotted in Green when it is above 60 and plotted in Red when it is below 40. Between 40 and 60 the RSI is plotted in Blue.

Markers indicate the state of each study.   A green ball is shown when a study is in its Buy zone. A red ball is shown when a study is in its Sell zone.   Since the RSI study has 3 states, a yellow ball is shown when the RSI is between its Buy zone and its Sell zone.

The Buy Signal is the consensus of all 3 studies being in their Buy zone.   The Sell Signal is the consensus of all 3 studies being in their Sell zone.

A-D – Variables are assigned the Boolean flag results of the study relationships.

E-F – Manage the color of the marker for the Average study on the 90% grid location.

G-H – Manage the color of the marker for the SMI study on the 70% grid location.

I-J – Manage the color of the marker for the RSI study on the 50% grid location.

A 2nd DYO tests for the consensus for the Buy and Sell signals.

Study Name	Study Location	Marker Location	Label Location	Grid Tab		Use as Default
Study Consensus 2	Chart	25% Grid	Column 7	Default		Draw Behind Bars
Study Scale			Variables File	Message Location		Privatize
Chart Scale			-Consensus	None		Close Only
						Show Values
Message Text		Font True False Panel				Email
		■ █ █ █				Auto Remove
						Auto Completion

Sound					Trading System			
◉ Silent	○ Beep	○ Voice	○ WAV	☐ Sound once per bar	Price		Quantity	Commission
WAV file					Last		1	0
			Browse					

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
Action	0	= if ## then marker size Offse		4	2	✓	🟢	🟩
		([BuyFlag]			0	) BUY		

A	Expression	[BuyFlag] := ([Ave>] AND [SMI>]) AND [RSI>60]	☐				
B	Expression	[SellFlag] := [RSI<40] AND Not ([Ave>] OR [SMI>])	☐				
C	Action	if ([BuyFlag]) then marker size 2 in column 4	✓	🟢	🟩	25% Grid	BUY
D	Action	if ([SellFlag]) then marker size 2 in column 4	✓	🔴	🟥	25% Grid	SELL
E	Action	if Not ([BuyFlag] OR [SellFlag]) then marker size 2 in column 4	✓	🟡	🟦	25% Grid	Wait

A – The Buy signal is True when the 3 studies are in their Buy zones.

B – The Sell signal is True when the 3 studies are in their Sell zones.

C-E – Manage the color of the marker and the message for the Buy and Sell signals.

The markers use the large 32 pixel icons by setting the Offset field to a value of 2. The column location for the marker is entered in the Selection #2 field, while the location for the Label is obtained from the Label Location selection. See Marker Size.

Though the example illustrates consensus for 3 studies, it could be expanded to evaluate more studies and show more markers.

Study Crossing Alert

The example demonstrates adding an audio alert when a study condition exits. The condition chosen was when the two lines of a Moving Average cross. The chart contains a Moving Average and a DYO.

A – Selects a Study flag for the Average study. The flag is 1st line X<> 2nd Line.   The 'X<>' notation means 'cross above or cross below'.

B – When the flag from Row A is True, the DYO will play the sound in the Sound frame. The example shows the Voice bullet is selected, which will read the text from the Study Name field. The words 'Study Crossing Alert' will be spoken.

Notation	Meaning
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
X>	Cross above
X<	Cross below
X<>	Cross above or cross below

Note: Check the Sound Once Per Bar box to have the alert sound once per bar instead of with every tick.

Tillson T3 Average

The Tillson template implements the Tillson T3 moving average formula.

Formula

    e1 = Mov(C,3,E)
    e2 = Mov(e1,3,E)
    e3 = Mov(e2,3,E)
    e4 = Mov(e3,3,E)
    e5 = Mov(e4,3,E)
    e6 = Mov(e5,3,E)
    c1 = -.618*.618*.618
    c2 = 3*.618*.618+3*.618*.618*.618
    c3 = -6*.618*.618-3*.618-3*.618*.618*.618
    c4 = 1+3*.618+.618*.618*.618+3*.618*.618
    T3 = c1*e6+c2*e5+c3*e4+c4*e3

	Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
	Expression ▾	166 Part 1 ▾	= -0.23603 ▾	* ▾	[165 Ave6] ▾	0 ▾
	🗗 🗗 🗗	+ ▾	(1.8539 ▾	* ▾	[164 Ave5] ▾	0 ▾)
A	Function	[160 Ave1] := Exponential Ave(Close, 3)			☐	
B	Function	[161 Ave2] := Exponential Ave([A], 3)			☐	
C	Function	[162 Ave3] := Exponential Ave([B], 3)			☐	
D	Function	[163 Ave4] := Exponential Ave([C], 3)			☐	
E	Function	[164 Ave5] := Exponential Ave([D], 3)			☐	
F	Function	[165 Ave6] := Exponential Ave([E], 3)			☐	
G	Expression	[166 Part 1] := (-0.23603 * [165 Ave6]) + (1.8539 * [164 Ave5])			☐	
H	Expression	[167 Part 2] := (-4.85363 * [163 Ave4]) + (4.2358 * [162 Ave3])			☐	
I	Expression	[168 Tillson T3] := [166 Part 1] + [167 Part 2]			☑ ⌒ ■	Study Value

A – Calculate an Exponential average of the Close using a period of 3.

B – Calculate an Exponential average of Row A's average.

C – Calculate an Exponential average of Row B's average.

D – Calculate an Exponential average of Row C's average.

E – Calculate an Exponential average of Row D's average.

F – Calculate an Exponential average of Row E's average.

G – Weight the averages from Rows E and F and sum them. The coefficients shown in the original formula have been multiplied out and entered as values shown in the DYO.

H – Weight the averages from Rows C and D and sum them.

I – The Tillson T3 is the sum of the parts. The T3 average is plotted on the chart.

Trade Runner System

The Trade Runner System template is documented in the DYO manual.

Weekly High and Low

This example will discover the weekly high and low values from the bars on the chart.

A - Set a Variable to a flag which is True on the 1st bar of the new week.

B-C - On the 1st bar of the new week, set the Current weekly values in variables as the Prior weekly values.

D - On the 1st bar of the new week, reset a bar counter to zero.

E - Increment the bar counter variable. It counts the number of bars in the week.

F-G - These statements find the Highest High and the Lowest Low in a set of bar values. The size of the set is the bar counter. The set includes the current bar and bars ahead of the current bar. Blue lines plot these values. The value in Op.[#] is 229, which is the Index for the BarCount variable.

Values in the range of 201 through 230 in this field are indexes to Variables.   The variable BarCount holds the value used as the function parameter.

The previous example could be accomplished using another approach that does not use a bar counter and the functions for Highest and Lowest.   Here is an alternate method, which should be easier to understand.

	Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	
	Function	233 CurrentWee	= Maximum(#2, ##)		High	0	
			([CurrentWeekHigh]			0	
A	Flag	[NewWeek] := 1st bar of New Week					
B	Action	if ([NewWeek]) then [PriorWeekHigh] := [CurrentWeekHigh]					
C	Action	if ([NewWeek]) then [PriorWeekLow] := [CurrentWeekLow]					
D	Action	if ([NewWeek]) then [CurrentWeekHigh] := -9999999					
E	Action	if ([NewWeek]) then [CurrentWeekLow] := 9999999					
F	Function	[CurrentWeekHigh] := Maximum(High, ([CurrentWeekHigh]))	✓ ⌐ ■				Study Value
G	Function	[CurrentWeekLow] := Minimum(Low, ([CurrentWeekLow]))	✓ ⌐ ■				Study Value
H	Expression	[0] := [PriorWeekHigh]	✓ ⌐ ■				Study Value
I	Expression	[0] := [PriorWeekLow]	✓ ⌐ ■				Study Value

A-C - These rows are the same as in the prior example. Prior week variables are set from the Current week variables before the Current week variables are reset on the 1st bar of the new week.

D-E - Initialize the Current Weekly High variable to a very low value. Initialize the Current Weekly Low variable to a very high value. This initialization is done on the 1st bar of the new week.

F-G - These functions compare current bar values with the weekly variables and set the weekly variables. If the bar's High is higher then the Current Weekly High, then the High is returned and assigned to the Current Weekly High variable. The Maximum function compares two parameters and returns the higher value.

H-I - These lines plot the values saved in the Prior weekly variables.

William's Variable Accumulation Distribution

The Williams Variable Accumulation Distribution (WVAD), developed by Larry Williams, is a volume-weighted price momentum indicator. It measures the buying and selling pressure by calculating the relationship between the number of points the market has moved from the open to close relative to the period's entire range. Short positions are taken for a negative moving average value, and Long positions for positive average values.
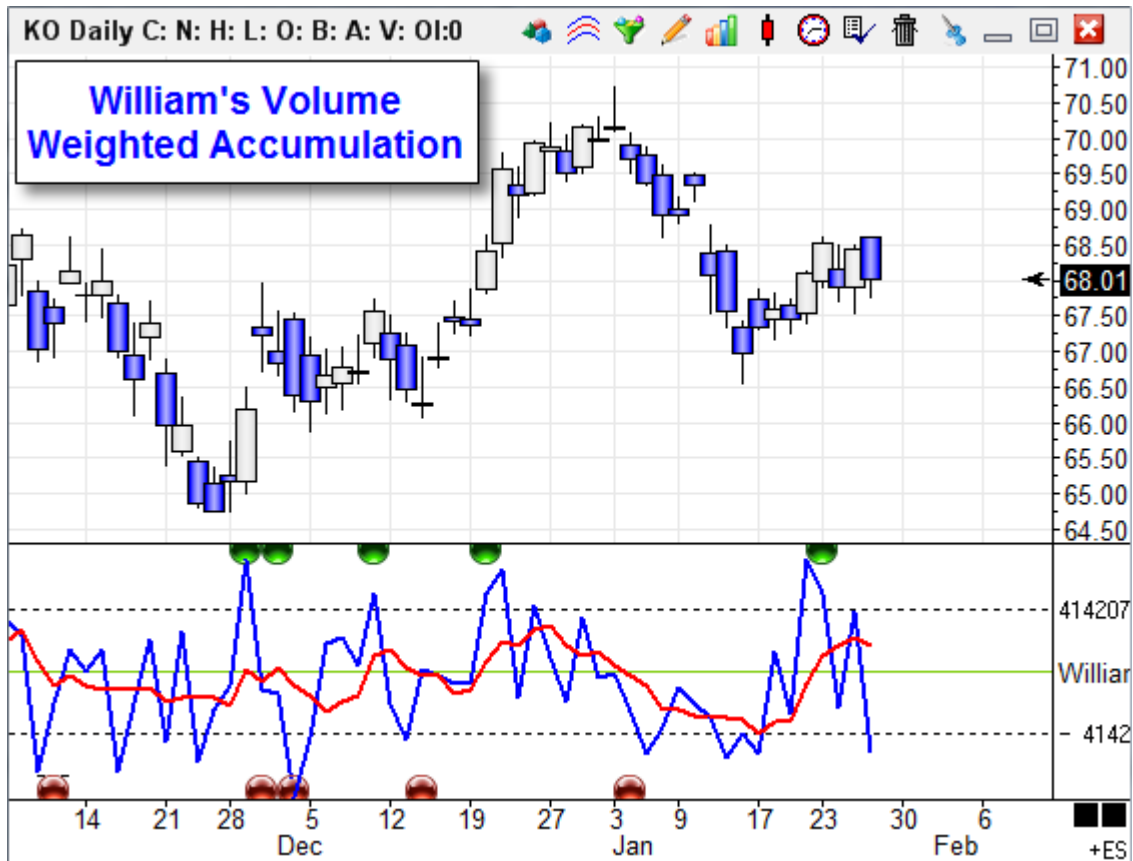
Formula

WVAD = ((Close – Open) / (High – Low)) * Volume

The formula can be implemented with a DYO, as shown.   This is available as a package to download from the Ensign web site using the Package feature in Ensign 10.

	Category	Selection	
A	Expression	[Period] := 5	
B	Expression	[WVAD] := (C-O)/(H-L) * Volume	
C	Action	[0] := [WVAD]; if (True) then plot V	Study Value
D	Function	[SMA] := Simple Average([B], [Period])	Study Value
E			
F	Expression	[Long] := [D] X> 0	Top Row 1
G	Expression	[Short] := [D] X< 0	Bottom Row

A – Period for Moving Average (Line D)

B – Implements the formula.   WVAD = ((Close – Open) / (High – Low)) * Volume

C – Plot the formula values. This statement is used because it will plot values that equal zero, whereas plotting directly from line B would not.

D – Simple Moving Average of the WVAD. Plotted in Red.

F – Mark where the average crosses above zero. This is the Long signal.

G – Mark where the average crosses below zero. This is the Short signal.

Woodies CCI

A WoodiesCCI template can be downloaded from the Ensign web site using the Package feature in Ensign 10. The WoodiesCCI template dresses a chart with the following indicators.

The template includes the following pieces of information. See the corresponding number labels on the chart.

1. An alert message shows at the top of the chart when the CCI line is above 200 or below -200.
2. 34 EMA study
3. 25 LSMA study
4. Candlesticks, Ensign Rockets, Ensign Zebras, or Ensign Flutes. Chart shows Solid Candles.
5. Big yellow block markers showing past CCI extremes above 200 and below -200.
6. Small yellow pegs are shown for the first occurrence of a higher high at the top of the sub-window, and for the first occurrence of a lower low at the bottom of the sub-window.
7. Day Session High value and the Net to the current price from this High.
8. +200 grid line colored green when Close is above the 34 EMA and colored red when below.
9. The Daily High-Low Range.
10. High of prior bar and current bar.   Current price. Low of prior bar and current bar.
     The High values will show in green when the current bar has a higher high.

The Low values will show in red when the current bar has a lower low.

11. Number of ticks in the current bar.
    Angle in degrees of the slope of the LSMA study line.
    Angle in degrees of the slope of the EMA study line.
12. Total of the 2 angle values. Trend labels of Flat, Normal and Trending based on the Total value.
13. Day Session Low value and the Net to the current price from this Low.
14. -200 grid line colored red when the angle sum is Flat, yellow when Normal, and Green when Trending.
15. 0 grid line colored green when Close is above the 25 LSMA and colored red when below.
16. Primary 14 period CCI study indicator. Plotted as a thick black line.
17. Fast 6 period CCI leading indicator. Plotted as a thin yellow line.
18. Histogram colored gray after crossing the zero line, yellow on the 5th bar, and red or blue when trending.

Explanation

Ken Wood (aka 'Woodie') typically does not show the bar chart portion. Instead he watches just the CCI study which is a sub-window in our presentation. However, some of the values being referenced are better shown on the bar chart, so for the sake of illustration, the chart has been shown with the study sub-window.

On the example chart is an Exponential Moving Average (EMA), and its parameter is 34. This is the blue and pink line (#2) using the Above | Below format. The Above | Below is comparing the 34 line to a 9 period 2nd line which is not shown. The 2nd study line (#3) on the chart is a Linear Regression Moving Average (LSMA) with a 25 period parameter. This line is being plotted in Green and Red using the Rising | Falling format. References to 34 EMA and 25 LSMA are referring to these 2 study lines.

Woodie follows a CCI spread line which is plotted in the sub-window as the thick black line (#16), and it oscillates above and below the zero center line. A histogram style (#18) is used with this CCI line with 4 different colors for the histogram bars. When the CCI line crosses zero line, as it did at 8:00 in the example, the histogram changes to a dark gray to indicate the crossing. If the CCI is still on the same side of the zero line 5 bars later, the gray color changes to a yellow color to indicate being at the threshold of a trend. Subsequent bars after the yellow boundary bar will be colored red when below the zero line, indicating trending, or blue when above the zero line and trending.

This presentation of the CCI with the colored histogram in Ensign matches the visual presentation Woodie uses. This primary CCI line in black uses a 14 parameter, and in Ensign it uses a 67 multiplier parameter. A 2nd CCI line (#17) is plotted as a thin yellow line to indicate a leading CCI which gives clues of what change might be forthcoming in the 14 CCI line.

To the right of the zero grid line for the CCI is the current price. Above this current price are a pair of prices (#10), and these 2 prices are the values of the prior bar's high and the current bar's high. The template will change the color of these 2 prices to be a green color when the current bar has a higher high. Below the current price are similar prices for the prior bar's low and the current bar's low.   And as shown in the example, these prices, normally shown in white, will change to be shown in red when the current bar has a lower low. So, in the center of the window one can quickly see the current price in the range of the prior bar and the current bar, and whether the current bar is breaking higher or lower than the prior bar.

Items #12 and #14 on the chart are Ensign's equivalent of Woodie's Sidewinder indicator. Woodie has a trending indicator that is watching the slope of the 34 EMA and the slope of the 25 LSMA. He calculates an angle from these slopes and sums the two angle values to get an indicator for trendiness. I have looked over the details of the math he used to get his angle information, and I consider the Pyrapoint angle technology we have in Ensign to be similar enough to be used as an effective substitute.

And it is a substitute for 2 reasons....1) I consider the Pyrapoint angle math to be superior and more generalized at arriving at a consistent angle regardless of the chart time frame or symbol the tool is applied to. Woodie's angle calculations are dependent on the chart time frame, etc., which is fine, but our method works very well also.

The 2nd reason for the substitution is that the math for our Pyrapoint angle calculation is built into the Ensign program so it computes very quickly, much faster than could be implemented with a lengthy DYO sequence to calculate the angle as per Woodie's method. I have watched and compared our angle values with Woodie's, and in general, our angle values tend to be smaller numbers. Therefore the -64 degrees for the 25 LSMA and the -2 for the 34 EMA might be smaller values than Woodie's calculations. These angles are added together, which at times cancels out some of the difference. The message of Flat, Normal, or Trending that is shown by the -66 degrees total value, comes from certain levels of the angle total.

The -200 grid line (#14) is a color representation of the message that was shown at that point in time for the bar. The red dots are shown when the angle total's absolute value is below 50 and thus the word 'Flat' is being displayed. 50 is the threshold I found to use that makes the dash coloring match quite well the same visual shown on Woodie's charts. Sums between 50 and 70 are the normal zone, and shown in yellow and would show the message label 'Normal' (#12). Angle totals that exceed 70 are shown in green and post the message 'Trending' in green.

Again, this is slightly different math, but still accomplishes an equivalent trending indicator. Ensign's study is equivalent to Woodie's Sidewinder because they look really similar as trending indicators and messaging. You will find the -200 grid line coloring to be very similar to Woodie's. The colors of Red, Yellow and green mean Flat, Normal and Trending in that order. You will find that the angles are slightly different for the 34 EMA and 25 LSMA on Ensign's charts from what you may see on the moderator's charts. Ensign uses its proprietary Pyrapoint technology to determine standardized angles for the study line slopes.

Most Woodie patrons don't look at prices because the tendency is to then focus on prices and not the study patterns. Do as Woodie says, let the patterns put you in a trade and take you out of the trade. Do not make up trades. Draw trend lines and use the information on the screen to confirm the patterns. Use both of the degree numbers to gauge direction and strength of the lama and mea. Use the total to gauge the strength of the CCI trend. Use the -200 grid line to keep a history, and use the line and label color to tell when to accept a ZLR trade.

For more information and documentation about Woodie's CCI trading methods, please visit his web site at www.woodiescciclub.com