

DYO Manual

Design Your Own

Ensign 10

Copyright © 2014 Ensign Software, Inc.

Last Update: 22 March 2014

Table of Contents

Overview	5
Add DYO to a Chart	5
DYO Property Form	5
Study Name	5
Study Location	5
Marker Location	6
Label Location	6
Grid Tab	7
Study Scale	7
Message Location	7
Row Logic	8
Message Text	8
Script Entry	9
Category	9
Variables	11
Variables File	11
Data Fields	11
Bar Offset	12
Operators	12
Show Marker and Label	13
Nomenclature	14
Form Controls	16
Fields View	16
ESPL Calls	17
Programming Tips	18
Examples	21
Net Price Marker	21
Marker Size	22
Paint Color	22
3-Bar Signal	23
DeMarker Study	25
Formula	25
ESPL Script	26
DYO Arrays	27
ESPL File	27
Study Consensus	28
Study Crossing Alert	31
Price and Time Bands	32
Show Marker	33
Hourly Bars	34
Encapsulation	36
Pesavento Break	37
Dynamic Support and Resistance	39
Formula	39
Dynamic Fibonacci Levels	41
Section Font Size	43

Label Font Size	44
Study Labels	45
Weekly High and Low	46
Draw Statements	48
Data Point	48
MoveTo	48
LineTo	48
HorzLine	48
VertLine	49
Rectangle	49
Circle	50
Ellipse	50
Arc	50
Move Alan Square	51
Move Fibonacci Levels	54
Move Andrews Pitchfork	55
Refresh Button	56
Scoreboard	57
Spreadsheet Statements	59
Trading System	61
Optimize Trades	64
Iterations	64
Execute Script	65
Results	66
File of Optimal Parameters	66
Optimizer Running	67
Summary	67
Trend Runner System	68
Study Settings	69
Trades	70
Profit Curve	71
At Risk	72
Calculate Stats	73
Show Stats	73
Fill Markers	74
Trade Markers	75
Email Notifications	75
Move SELL Stop	76
Move BUY Stop	76
Profit Target	77
BUY/SELL Entry	77
Optimizer	78
Trade Report	80
Parameter Automation	81
Template	81
Chart Scanner	82
Symbol List	82
Criteria	83
Report	83

<u>Script</u>	84
<u>Results</u>	84
<u>Open a Chart</u>	85
<u>Sorting</u>	85
<u>Save to File</u>	85
<u>DYO Setup</u>	86
<u>Show # with Variable name</u>	86
<u>Send to Application Title</u>	87
<u>Appendix</u>	88
<u>Expression Data Values</u>	88
<u>Key Pressed</u>	90
<u>Operators</u>	91
<u>Function Category</u>	91
<u>Flag Category</u>	93
<u>Action Category</u>	94
<u>Price Finder</u>	97
<u>V := # ; plot V as study transfer</u>	97
<u>Row #2 Offset[1] := ##</u>	97
<u>Action Statements involving Strings</u>	97
<u>if ## then Speak(#2)</u>	97
<u>if # then Output(Message)</u>	
<u>if ## then Output(#2)</u>	98
<u>if # then Clipboard(Message)</u>	
<u>if ## then Clipboard(#2)</u>	98
<u>If # then Alert(Message)</u>	
<u>If # then Alert(#2)</u>	98
<u>If # then Send(Message)</u>	99
<u>If ## then Send([#], #2)</u>	99
<u>if # then SendKeys(Message)</u>	99
<u>if # then AppendFile(Message)</u>	99
<u>if ## then AppendFile(#2)</u>	99
<u>if #2 then V := ReadFile(#3, #4)</u>	100
<u>If # then E-mail Message</u>	
<u>If # then Text Message</u>	101
<u>If # then Action #2</u>	102
<u>Study Category</u>	103
<u>Quote Value Category</u>	104
<u>Chart Value Category</u>	105
<u>Branching Category</u>	106
<u>ESPL and DLL Categories</u>	106
<u>Reference Tags</u>	107
<u>Formatting</u>	108
<u>Markers</u>	109
<u>Image List</u>	113

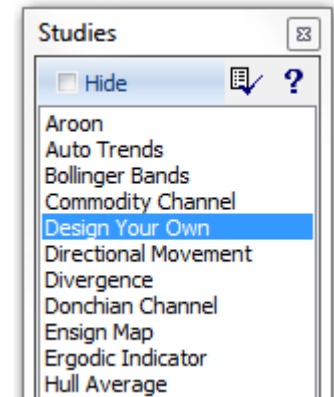
Design Your Own (DYO)

Overview

The Design Your Own (DYO) study feature can be used to implement Boolean logic and mathematical equations. The feature can be used to create and plot custom indicators, or show special visual effects. The variety and complexity of the things being created by customers with the DYO feature are truly amazing.

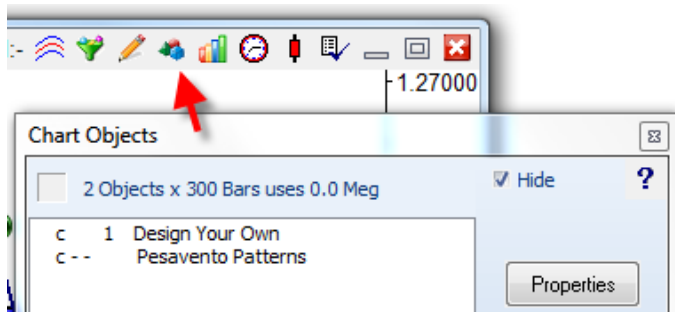
Add DYO to a Chart

With a chart showing, click the Study button and select Design Your Own on the drop down list. This adds a DYO study to the chart's collection of studies, alerts and draw tools.



DYO Property Form

Click the Chart Objects button to show the Chart Objects list. Double click on the Design Your Own entry on the list.



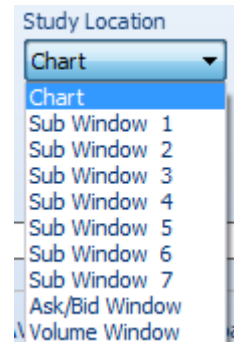
The DYO property form will be displayed so properties can be set.

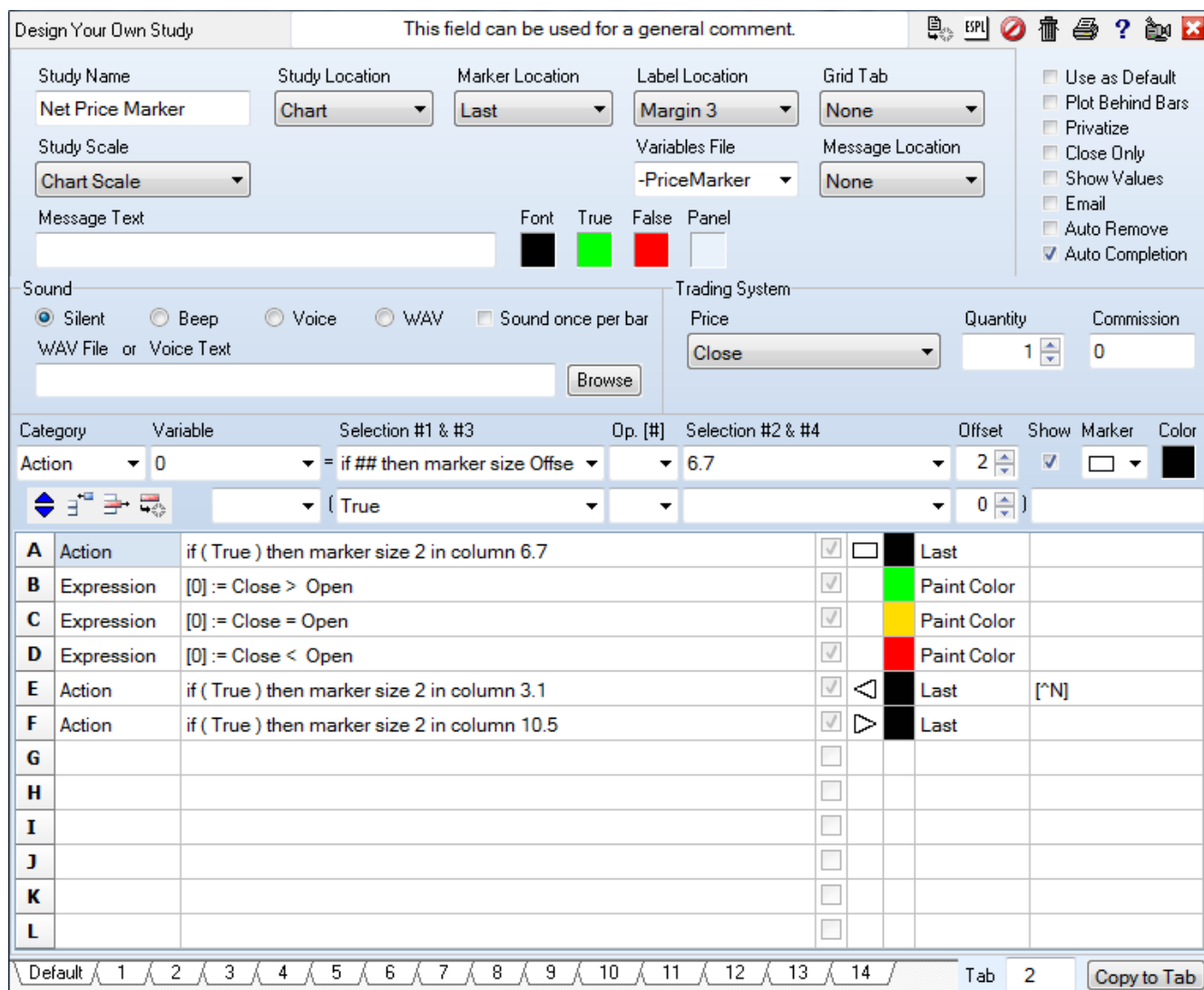
Study Name

Change the Study Name field to give your DYO a custom name. This name will show on the Chart Objects list and be used as the label in a sub-window.

Study Location

This selects the panel where the Markers and Labels will be displayed.





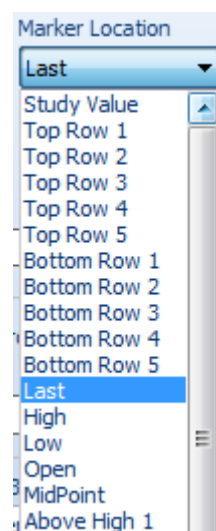
Marker Location

Each DYO row can have its own Marker, Color, Marker Location, and Label. The Marker Location selection would be Study Value to plot a curve at its calculated values.

The other positions can be used to place markers on the chart or sub-window at a particular position, such as in rows along the Top or Bottom of the panel, Above or Below the High and Low of a bar, or at vertical locations on the scale. The selections of Last, High, Low, Open, and Midpoint will use that price position on a bar.

Label Location

The Label Location controls the horizontal position of the Label text that can be shown in the right side margin of the chart. Column 0 begins at the center of the last bar on the chart, and each column is 8 pixels to the right.



Grid Tab

A DY0 that is placed in a sub-window can establish the grid lines shown in the sub-window. The selection is the tab from the Grid object.

Study Scale

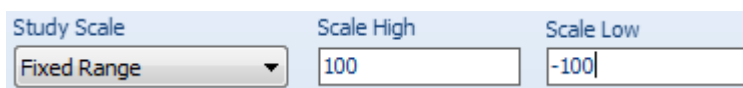
Use the Study Scale selection to pick the scale that is appropriate for the data to be plotted.

Chart prices plotted on the chart would typically use Chart Scale. An example would be the plotting of a moving average study.

Data plotted in an existing sub-window would typically use the sub-window's scale.

The Data Set selection will determine the scale from the data by looking for the highest high and the lowest low value in the data set. This is a common selection for plotting studies from user created formulas.

The Fixed Range selection will show additional boxes to enter a custom scale.



The image shows a control panel for 'Study Scale'. It contains three main components: a dropdown menu labeled 'Study Scale' with 'Fixed Range' selected, a text input field labeled 'Scale High' containing the value '100', and another text input field labeled 'Scale Low' containing the value '-100'.

Data 0 Centerline will balance the scale so zero plots at the center of the panel.

Data 50 Centerline will balance the scale so 50 plots at the center of the panel. This is the selection used in the Stochastic example because this study moves on both sides of 50 in a range with zero as the minimum value and 100 as the maximum value.

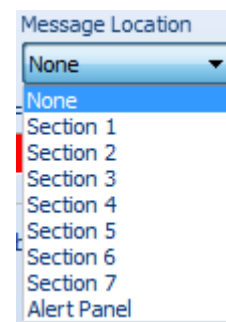
0 to Max Data will use zero as the scale low and determine scale high from the data set. This would be a good selection for plotting histogram bars for volume values.

The last four selections are similar to those described. The scale range is increased to create a 5 percent margin at both the top and bottom of the range.

Message Location

Ensign has a sub-window at the bottom of the chart that is divided into seven Sections. This area can be used to show messages, study and chart values.

The DY0 can display a message in one of the seven Sections. Select None to disable the Message feature.



The image shows a dropdown menu titled 'Message Location'. The menu is open, displaying a list of options: 'None', 'Section 1', 'Section 2', 'Section 3', 'Section 4', 'Section 5', 'Section 6', 'Section 7', and 'Alert Panel'. The 'None' option at the top is currently selected.

Row Logic

The Row Logic selection identifies which DYO row controls the Section background color. When the Row Logic is True, the section background will use the True color. When the Row Logic is False, the background will be colored the False color. The message text will show in the Font color.

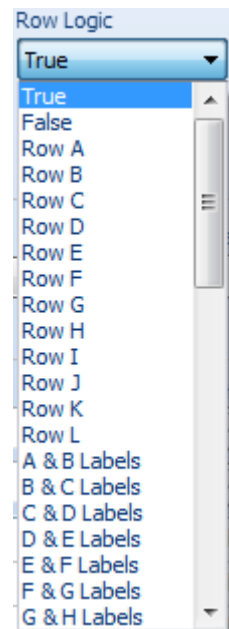
Row A through Row L can be a Boolean result that will control the Section color and the message is from the Message Text edit box.

Selections 'A & B Labels' through 'K & L Labels' will use the Boolean logic of the two rows as a pair. Which ever row of the pair was most recently True is the row whose label will be displayed in the Section as the message text. The Section background color will be the row's Show color.

'A Label True' through 'L Label True' will show the label from these rows only when the row evaluates to a true. When the row is a False, nothing is done.

'A or B or Blank' through 'K or L or Blank' will show the row's label when either row evaluates to True and use the row's Show color. When both rows are False, the Section is Blank and colored using the False color.

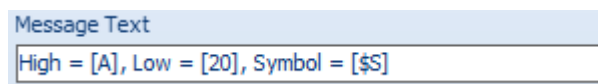
'A or B or C' through 'J or K or L' will show the row's label when one of the rows evaluates to True and use the row's Show color.



Message Text

The Message Text can be shown in a Section at the bottom of a chart. Text to be shown on multiple lines is separated by a comma character.

The text can contain a message to show when the Row Logic is True and a message to show when the Row Logic is False. Separate the two messages with a vertical line character.



The example illustrates a 3 line message with insertions at the [.] reference tags. [A] through [K] will insert the DYO's row value. The example shows the value from row A by using [A].

[0] through [999] will insert a Variable value. The example shows the value in Variable twenty by using [20].

Boolean flags will show 'True' or 'False'.

Reference Tags - Chart and Bar values can be referenced and inserted in the Message Text, a Label, and used in an expression field to return a value. The Message Text example used [\$\$] to print the chart's Symbol in the Section display.

A short list of tags is shown in this table. [Appendix – Reference Tags](#) has the full list.

[\$\$]	Chart symbol
[\$C], [\$H], [\$L], [\$O]	Bar Close, High, Low, Open
[\$V], [\$T]	Bar Volume, Tick Count
[\$N], [\$R]	Bar Net, Range
[\$A], [\$B]	Ask Volume, Bid Volume

Formatting -

Character	Description	Example
\$	Add a \$ suffix to format the value as a price using the chart's scale.	[A\$]
.	Add a period suffix to format the value as an integer.	[\$V.]
.#	Add a period suffix and the number of decimals to show.	[A.2]
`	The ` will add a degree symbol. ` shares the ~ key and follows the [].	[A]`

Script Entry

Use the DYO rows to implement logic and math operations. The rows will be evaluated in sequence from row A through row L. When more rows are needed to complete a design, simply add more DYO studies to the chart and continue the implementation.

Category

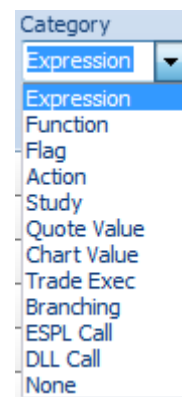
Several hundred DYO statements have been grouped into 12 Categories. The first step to implementing a design is to select a category on a row, which then displays appropriate additional entry fields.

Expression - This category is used to write both Boolean and algebraic expressions by combining data values, Variables and DYO row values.

Function - This category contains 46 mathematical functions such as Sqrt for Square Root. [Appendix – Function](#)

Flag - Look in this category for Boolean logic statements testing complex relationships between values, and operations with other Boolean flags. [Appendix – Flag](#)

Action - This category contains statements for manipulating Variables, complex IF statements, and controlling various actions in the Ensign program. [Appendix – Action](#)



Study - The values and flags from studies that are present on a chart ahead of the DYO on the chart's object list can be read. Select the Study from the Selection drop-down list. Select the particular study's value from the #2 drop-down list. For example, you could select the 'Stochastic' study from the Selection #1 list, and then select the '%D Value' from the Selection #2 list.

Studies that are placed on the chart will report a variety of conditions. These conditions are Boolean flags that can be used in an implementation. An example for the Stochastic study would be the flag for '%D Rising'.

[Appendix – Study](#)

Quote Value - This category can be used to read any of the values and flags from the quote page. Leave the Selection field blank to have the row default to the chart's symbol. Enter a Symbol in the Selection field to read values for some other symbol. Since values are obtained in real-time, the values will be for the content presently on the quote page. There is no back history for what the quote page values might have been in the past.

[Appendix – Quote Value](#)

Chart Value - This category list statements that deal with values and flags obtained from the chart such as whether a bar's time stamp is in the Globex session.

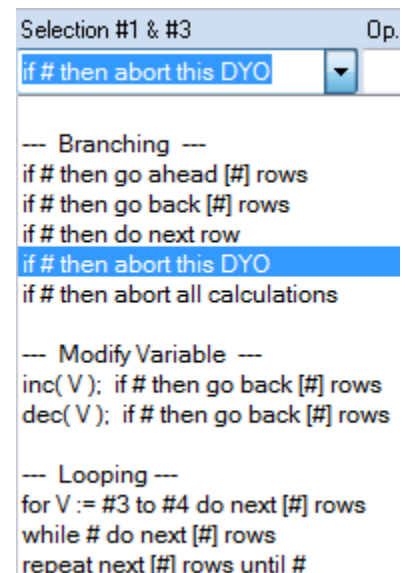
[Appendix – Chart Value](#)

Branching - The statements in this category implement conditional branching of program execution to another DYO row. Multiple rows can be skipped or branching can loop back to an earlier DYO row.

[Appendix – Branching](#)

ESPL and DLL Call – The DYO row can execute a function or procedure written in the ESPL language or make an external call to a DLL procedure.

[Appendix – ESPL and DLL](#)



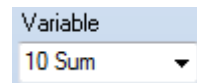
None - Use this category selection when a DYO row is not used. This selection hides the selection boxes for the row. When the DYO is evaluated, a row with a None category selection is skipped. A row that is changed to None will show in the table with a yellow background color. This is like commenting out the line. The category can be reselected to undo this comment mode. Use the Erase Row button to erase a row, which also sets the category to the None selection.

Variables

1000 Variables are available to hold values or Boolean flags. Variables with indexes in the range of 0 ..199, and 400..999 have a public scope so they can be written and read by studies on other charts. Variables with indexes in the range of 200 .. 399 have a private scope, meaning they can only be written and read by the current chart.

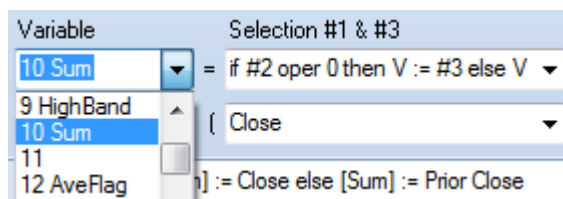
Variables can be used in expression fields, message and label text by surrounding the variable index with square brackets. Example: [10] refers to Variable number 10.

Variables can be given names to make the script more readable. In the Variable edit box, enter a new name to replace the index number. The text in the Variable edit box will show the index and the new name. But when a variable is used in an expression, the new name will show instead of the index number. Now the variable [Sum] can be used in expressions instead of [10].



Custom variable names can be returned to their index number by erasing the text in the edit box, or by double clicking on the Variable edit box when the variable has been selected.

Example: '10 Sum' shows in the Variable field, and a double click on the field will replace '10 Sum' with its index number, ie. '10'. In expressions, [Sum] will be replaced with [10].



Variables File

The names for the Variables are saved in the C:\Ensign10\Variables folder in a file with the name shown in the Variables File list box. The extension on the filename will be .ini. The file is ASCII and can be edited using Notepad.

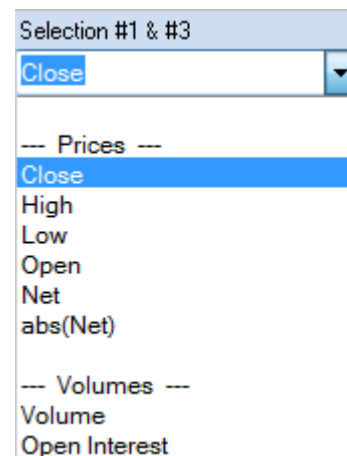
A new file is created by editing the text in the Variables File combo box. An existing file can be selected from the drop down list, which will change the name assignments for the variables. If you distribute templates and layouts containing DYO's that have variable names, please create a custom Variables File and distribute your personalized file. Save your file in C:\Ensign10\Variables folder and then the new filename will show in the combo box list.

Data Fields

Over 190 predefined data fields are available for use in writing expressions. The choices populate a selection drop-down list so it is easy to know what is offered. A selection such as 'Close' will return a bar's close value.

Lines with '---' are section headers for organizational purposes rather than data fields to be selected. Do not select these entries.

[Appendix – Expression](#)



Bar Offset

A Bar Offset can be specified to reference earlier bar data. For example, a Bar Offset value of -1 would read data from the prior bar. A value of -5 would indicate 5 bars back. An offset of zero is the current bar begin evaluated. Example: Close[-1] is the prior bar's close.

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Expression	0	= Close			0
	>	[Close			-5

A Expression [0] := Close > Close[-5]

A Variable with an index in the range of 201 to 230 can be the source of the value for the Offset. Enter the Variable's index in the Offset field.

Operators

Use the Op. list box to select mathematic and logic operators.

The '>' operator means 'greater than' and the Boolean flag is a relationship that could be True on several bars in sequence. The 'X>' operator means 'cross above' and the Boolean flag would be True on a single bar as an event.

Operator	Meaning	Type
>, >=	Greater than, Greater than or equal to	Relationship
=, <>	Equal to, Not Equal	Relationship
<, <=	Less than, Less than or equal to	Relationship
X>, X<	Cross above, Cross below	Event
X<>	Cross in either direction	Event

Op. [#]

- |
- +
-
- *
- /
- ^
- * -
- / -
- div
- mod
- AND
- AND Not
- OR
- XOR
- Not
- >=
- >
- =
- <>
- <
- <=
- X>
- X<
- X<>

[Appendix – Operators](#)

Show Marker and Label

Each DY0 row can show a Marker and/or Label in the selected color. When the Show box is checked, category statements that generate a value will always plot the Marker or Label. But category statements that return a Boolean flag will only show the Marker or Label when the Boolean result is True. (Note: A Boolean True returns a value of 1, and a Boolean False returns a value of 0.)

Marker - There are 177 [Marker](#) selections to choose from. Select the 1st entry of None when you do not want to display a Marker. The 1st eight selections would be used for plotting a study curve.

Color - Click on the Color square to show the Color Dialog form to change the color selection. Some markers use two colors, such as the Rise/Fall curve markers. The 1st color is set on the Row with the selection, and the 2nd color is set on the next Row.

Some markers are icons which use multiple colors. They will be drawn as shown on the marker list which means their color cannot be changed.

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.

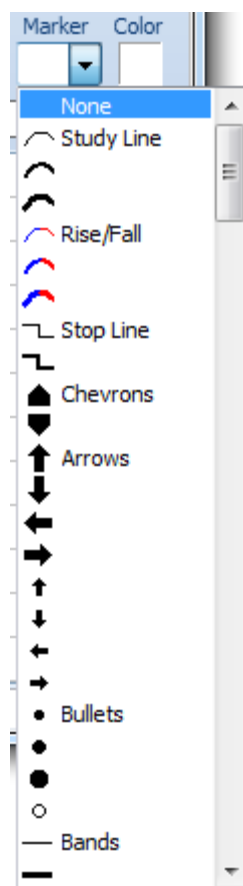
Label - The Label may have a combination of text and references tags as documented on the [Reference Tags](#) page. Labels will show in the margin to the right of the last bar on the chart at the Label Location position. Text to be shown on multiple rows is separated by a comma character. [A] through [L] will insert the DY0's row value. [0] through [999] will insert a Variable value.

The Label text can contain a prefix that specifies a Section window for the label. The prefix is in the format of |x| where x is the number of the Section window in the range of 1..7. Example: |1|Last=[\$C] will put Last=1325.50 in Section 1.

When showing a Label in a Section, if the row evaluates to True, the True color is used for the Section background. If False, the False color is used for the Section background. See the [Row Logic](#) documentation for setting these 2 colors.

Double click in the Label field to erase the label and reset the Show check box to being unchecked. The Marker is removed and the Color is reset to white.

There is limited space for text that is shared by the Study Name, Message Text, WAV file path and all Labels. If a label's text disappears, the solution is to shorten text elsewhere, such as study name. Labels do not need to have the prices entered, and a [A] reference may work just as well, and be a fewer number of characters. All text shares a storage space of 120 characters, which includes some non-visible delimiter characters.



Nomenclature

- Refers to the expression that can be formed using Selection #2, #3 and #4 and the 3 operators between these 3 fields. These fields are surrounded by the yellow frame in the following example. This example will increment Variable [63] if the expression is True.

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Action	63 Counter	= if # then inc(V)		Volume	0
		>		{ 1.5 * Volume	-1
A Action if Volume > (1.5 * Volume[-1]) then inc([63 Counter])					

- Several statements use the Selection #2 field. Therefore, **##** refers to the expression that can be formed using Selection #3 and #4 and the 2 operators on the bottom row. Example:

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Function	61 High	= Maximum(#2, ##)		Close	0
		{ 1.1	*	Prior Close	0
A Function [61 High] := Maximum(Close, (1.1 * Prior Close))					

The Maximum function evaluates two parameters and returns the larger value. The 1st parameter is the Selection #2 field, and the 2nd parameter is the expression in the fields surrounded by the yellow frame. The example saves the function result in Variable [61].

#[#] - Some statements need an integer parameter which is to be entered in the Op. [#] field on the top row. This is an override to this field's typical use as an operator. Example:

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Function	61 High	= Highest(#2, [#])	10	High	0
		{			0
A Function [61 High] := Highest(High, 10)					

The Highest function has 2 parameters. The 1st is the Selection #2 field, which in the example refers to a bar's Close. The 2nd parameter is the size of the data set and the value is obtained from the [#] field on the top row. The example will return the Highest High in the last 10 bars, including the current bar because the Offset value is 0. The result is saved in Variable [61].

The range of values that can be entered for [#] are 0 through 200. Use the statement Action | Row #2 Operator[1] := ## to change the [#] parameter to a calculated or Variable value. The #2 selection for this Action statement would be the row letter in brackets like [B].

A Variable with an index in the range of 201 to 230 can be the source of the value for the [#]. Enter the Variable's index in the Op.[#] field.

#2, #3, #4 – Refer to Selection fields. Selection #2 is the right side field on the top row. Selection #3 is the left side field on the bottom row. Selection #4 is the right side field on the bottom row. Example:









Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Action	41 Value	= if #2 oper 0 then V := #3 els	>	[21]	0
		{ Close		Prior Close	0
A Action if [21] > 0 then [41 Value] := Close else [41 Value] := Prior Close					

This Action statement evaluates Selection #2 by testing to see if its value is greater than zero, and when True the Selection #3 is assigned to the Variable, and when False the Selection #4 is assigned to the Variable. The example reads: if Variable [21] is greater than 0 then Variable [41 Value] is assigned the bar's Close else Variable [41 Value] is assigned the Prior bar's Close.

V – V indicates the selected Variable will be used in the script, as in the prior Action example of 'if #2 oper 0 then V := #3 else V := #4'.

oper – Refers to the Operator selected on the top row in the Op. field. See the last example.

Form Controls

	Fields	This button changes the Script view to show a view of the Field entries.
	Script	This button changes the Field view back to the Script view. Note: The DYO form can be widened to fully show a lengthy script.
	Reset	This button resets (clears) the form in preparation to enter a new script.
	Copy	This button moves down all DYO rows below the selected row. A copy of the selected row will be inserted. Row L contents will be lost.
	Delete	This button deletes the selected row. All rows below the selected row will be moved up, and Row L will be blank.
	Erase	This button erases the selected row. The category will be changed to None. The entry fields will be hidden and the row will be blank.
	ESPL	Open the ESPL editor for writing ESPL functions.
	Move	Rows can be rearranged by clicking down on a Row letter and dragging vertically to a new Row location. The drop position is indicated by two green arrows. Release the mouse button to finish a move operation. Use Copy and Move as a quick way to duplicate similar rows of logic.
Row	Insert	There are 2 ways to insert a blank row in the middle of a script. 1 st – Move a blank row, such as Row L, to the desired location. 2 nd – Copy a row and then double click the duplicated row to erase it.
Row	Select	Click on a Row to select it. A selected row shows a highlight. The row's content can now be entered or edited using the selection boxes. Begin a new entry by selecting the Category.
Category	Clear	Double click the Category list box to clear the entry selection fields.
Variable	Clear	Double click the Variable list box to remove a custom name.
Selection	Clear	Double click a Selection field or Operator field to clear its content.
Label	Clear	Double click the Label edit field to clear the Label, the Show check box, the Marker, and the Color box.

Fields View

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Action	0	= if ## then marker size Offse		6.7	2
		{ True			0
A	Action	1 if ## then marker size Offset in column #2	2 6.7	3 True	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Last
B	Expression	[0] := 1 Close > 2 Open			<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Paint Color

In the Fields View mode, small numbers in green precede the field entries. The operators are shown in red. This display mode is useful when documenting how the script was created.

ESPL Calls

Any DYO row may call a function written in the Ensign Software Programming Language (ESPL). The call is made using a selection in the ESPL Call category. Wisely choose a selection that will minimize the calculation burden. For example, call the ESPL function for just the visible bars when a calculation is not needed for those bars that are off the screen.

Parameters - The DYO call will pass 5 parameters to the ESPL function in this order.

Index	The bar's index in the range of 2 through Bar Count.
V	The value of the Variable referenced by the DYO row.
#2	The value from the Selection #2 field.
#3	The value from the Selection #3 field.
#4	The value from the Selection #4 field.

```

Selection #1 & #3
if Index in Visible Bars then

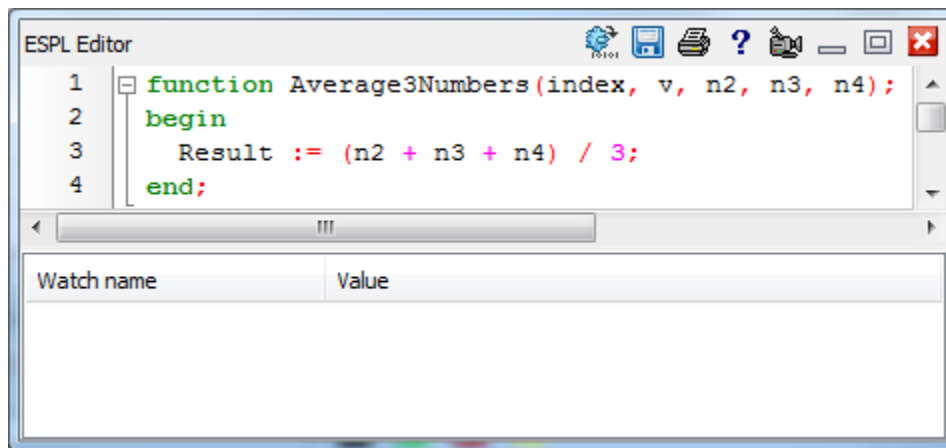
--- ESPL Function ---
V := Label ( Index, V, #2, #3, #4 )

--- Index Tests ---
if Index >= Bar Count - [#] then
if Index in Visible Bars then
if Index = 1st Visible Bar then
if Index = [#] then
if Index = #2 then

--- Boolean Tests ---
if #2 then
if # then
if ## then
    
```

Result - ESPL functions return their value in a variable named Result. The returned value is assigned to both the DYO row Variable and to the DYO row array.

Editor -  Click the ESPL button to open the ESPL Editor. Here is a simple example script.







```

ESPL Editor
1 function Average3Numbers(index, v, n2, n3, n4);
2 begin
3   Result := (n2 + n3 + n4) / 3;
4 end;
    
```

This function returns the average of the 3 Selection field values passed as parameters. See the ESPL documentation for a full description of the statements available for writing ESPL code.

Name - The name of the function to call is placed in the Label field of the DYO row. If the Label field is left blank, a default name made from double the DYO row letter will be used. Example: Call function BB for a call made from row B when row B does not have a label.

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
ESPL Call	0	= if Index in Visible Bars then		Close	0	<input checked="" type="checkbox"/>		
		[High		Low	0	<input checked="" type="checkbox"/>		Average3Numbers
A	ESPL Call	if Index in Visible Bars then [0] := Average3Numbers(Index, [0], Close,	<input checked="" type="checkbox"/>			Study Value		Average3Numbers

Programming Tips

Name – Use a descriptive Study Name because this name shows on the chart objects list.

Variables – Use variable names to make the script more readable and help document the logic. In the example which follows, the logic of Row A is self documented:

[BuyFlag] := ([Ave>] AND [Smi>]) AND [RSI>60]

Note that [RSI>60] is a variable name which self documents that it is a test for the RSI being above 60.

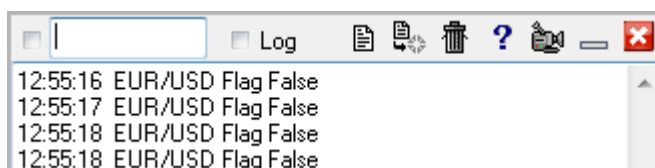
Variables Quote Page - The custom quote page named Variables shows the current values for all 400 variables. Variables with names show their names in the quote page cells.

Variable	0	1	2	3	4	5	6	7
0	True	Rule 1a=False	Rule 1b=False	Rule2=False	Rule3=False	DeMarker=False	Swing=False	Rising=False
10	False	False	AveFlag=True	SmiFlag=True	RSI>60=False	RSI<40=False	BuyFlag=False	SellFlag=False
20	Stochastic	False	False	LowDelta=False	SumHigh=False	SumLow=False	False	False

Trace – Use the Action | if # then Output(Message) statement to post variable values to the Output Windows. This statement can be an effective debugging aid to trace the flow of execution through DYO's. Example:

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show Marker	Color
Action	0	= if # then Output(Message)		True	0	<input type="checkbox"/>	
		(0	<input type="checkbox"/>	
A	Expression	[BuyFlag] := ([Ave>] AND [SMI>]) AND [RSI>60]				<input type="checkbox"/>	
B	Expression	[SellFlag] := [RSI<40] AND Not ([Ave>] OR [SMI>])				<input type="checkbox"/>	
C	Action	if ([BuyFlag]) then marker size 2 in column 4				<input checked="" type="checkbox"/>	25% Grid BUY
D	Action	if ([SellFlag]) then marker size 2 in column 4				<input checked="" type="checkbox"/>	25% Grid SELL
E	Action	if Not ([BuyFlag] OR [SellFlag]) then marker size 2 in column 4				<input checked="" type="checkbox"/>	25% Grid Wait
F						<input type="checkbox"/>	
G	Action	if True then Output('[\$\$] Flag [BuyFlag]')				<input type="checkbox"/>	
							[\$\$] Flag [BuyFlag]

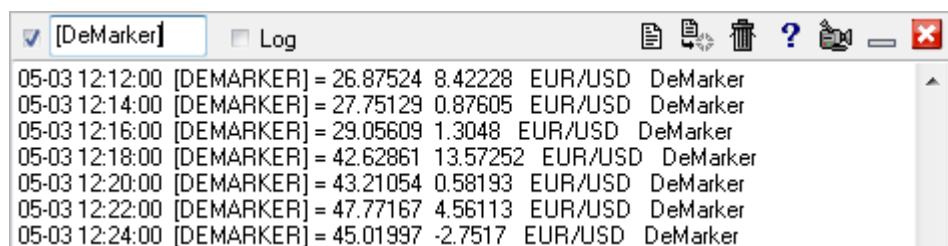
Every time this DYO executes, Row G will print in the Output Window the chart's symbol, the text 'Flag', and the value of the [BuyFlag] variable.



This is the Output Window display for this example. [BuyFlag] is a Boolean variable and a True is a value of 1.0. A Boolean False is a value of 0.0.

The (Message) part of the Output statement prints the contents of the Row's label. If the label is blank, then the content will switch to the DYO's Message Text line.

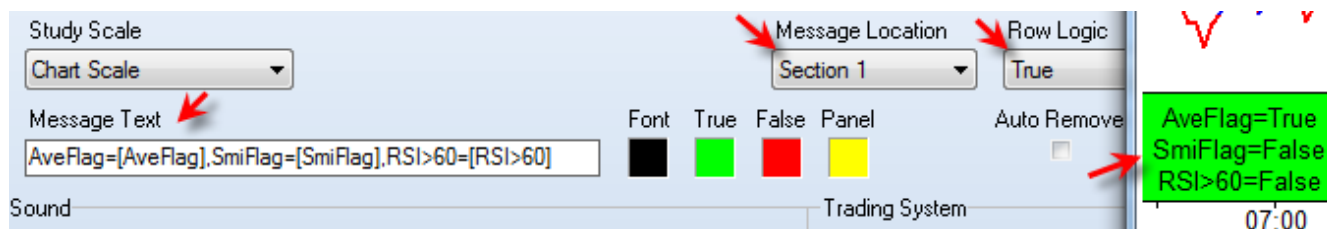
Debugger – A Variable can be entered as the keyword in the Output Window. This will trace the changes to the variable.



This example traces the contents of the [DeMarker] variable. The trace shows a time stamp, the variable name, its value, the change from its prior value, chart symbol, and the name of the study which changed the variable. This is a quick and easy way to trace a variable without making changes to a DYO to add the Action | if # then Output(Message) statement. Note that one or more Action statements can be use simultaneously with the Debugger variable, and all messages will be interleaved in the Output Window.

Uncheck the first check box to stop Debugger entries from being posted. Execution continues, however.

Section – The Section windows can be used to display multiple variables.

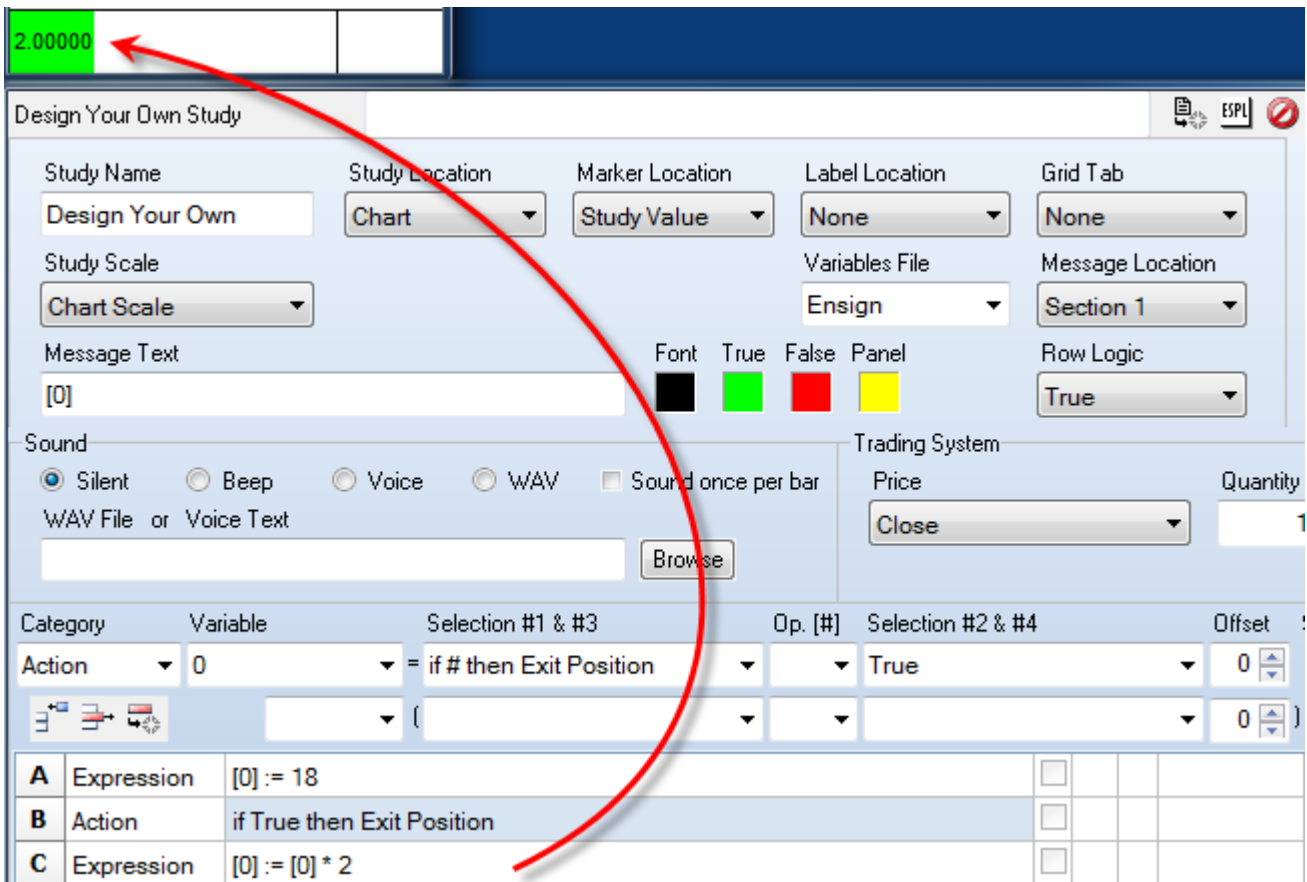


Markers – Use markers which show on the top or bottom row locations of the chart to show the state of Boolean expressions. The display of the markers to show when a Boolean flag is True can be temporary during development, and then removed by unchecking the Show box when the design is complete.

Print – Click the ribbon Window | Print button to print an image of the DYO property form so you have a paper copy of your DYO design.

Avoid Using [0] – All statements return a result which is assigned to the selected Variable, which by default is [0]. However, not all statements show this assignment in the script text that is displayed. The fact that a result of a Boolean test is probably being assigned to variable [0] is often overlooked.

The following example illustrates a reason to avoid using [0] as a variable for calculation results that will be needed later on in this DYO or in other DYOs.



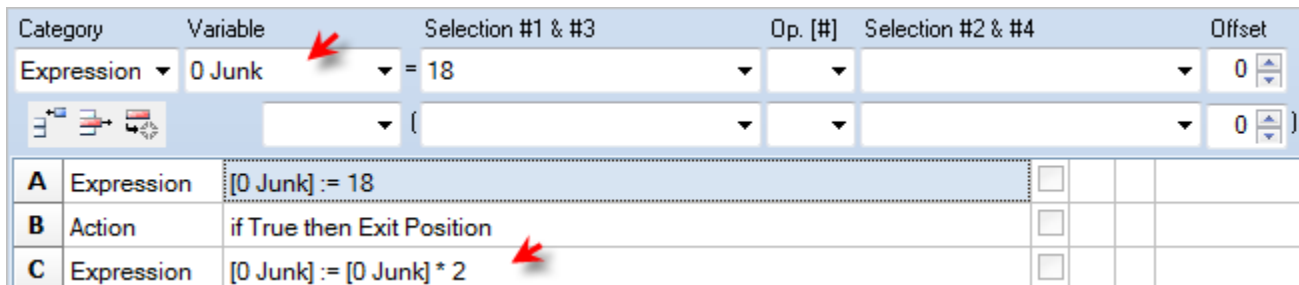
A – The expression assigns a value of 18 to variable [0].

B – This Action statement returns a Boolean result which is assigned to variable [0]. This assignment to [0] is not obvious because it is not shown in the script text on row B. After row B executes, [0] will have a value of 1 for True. A Boolean False result would assign a zero.

C – Variable [0] is used in a calculation. The Message Text shows variable [0] in the Section 1 Message Location. A value of 2 is shown when a value of 36 was expected. It is row B's Boolean result that replaced the contents of [0].

Tip: Do not use [0] to hold values that will be used elsewhere in the DYO's design. Row A made an assignment to [0], which was changed by row B before it was read on row C.

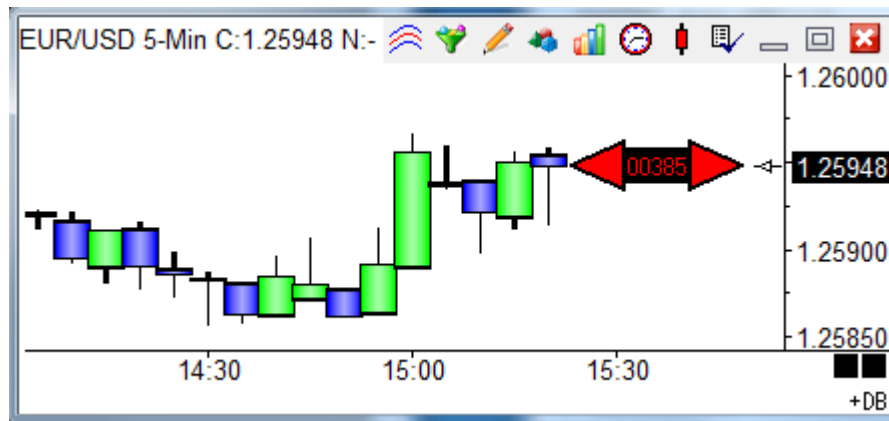
Tip: Assign the name 'Junk' to variable [0] as a reminder it should not be used for calculations. Implementing this tip would have given the following reminder on row C.



Examples

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. The marker will move up and down as ticks are received. The colors change from green to red to yellow depending on the net with the bar's open and the daily net value is shown in the body of the marker.



Study Name <input type="text" value="Net Price Marker"/>	Study Location <input type="text" value="Chart"/>	Marker Location <input type="text" value="Last"/>	Label Location <input type="text" value="Column 3"/>	Grid Tab <input type="text" value="None"/>	<input type="checkbox"/> Use as Default <input type="checkbox"/> Draw Behind Bars <input type="checkbox"/> Privatize <input type="checkbox"/> Close Only <input type="checkbox"/> Show Values <input type="checkbox"/> Email <input type="checkbox"/> Auto Remove <input type="checkbox"/> Auto Completion	
Study Scale <input type="text" value="Chart Scale"/>	Variables File <input type="text" value="Ensign"/>	Message Location <input type="text" value="None"/>				
Message Text <input type="text"/>		Font <input type="checkbox"/> True <input type="checkbox"/> False <input type="checkbox"/> Panel <input type="checkbox"/> True <input type="checkbox"/> False <input type="checkbox"/> Panel				
Sound <input checked="" type="radio"/> Silent <input type="radio"/> Beep <input type="radio"/> Voice <input type="radio"/> WAV <input type="checkbox"/> Sound once per bar WAV file <input type="text"/> <input type="button" value="Browse"/>		Trading System Price <input type="text" value="Last"/> Quantity <input type="text" value="1"/> Commission <input type="text" value="0"/>				

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
Action	0	= if ## then marker in column #2		6.5	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="button" value="Add"/> <input type="button" value="Remove"/> <input type="button" value="Reset"/>								
		(True			0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A	Action	if (True) then marker in column 6.5		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Last
B	Expression	[0] := Close > Open	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Paint Color
C	Expression	[0] := Close = Open	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Paint Color
D	Expression	[0] := Close < Open	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Paint Color
E	Action	if (True) then marker in column 3		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Last
F	Action	if (True) then marker in column 10		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	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 6.5, which is 52 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.

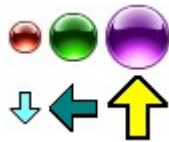
D – If the Close is less than the Open, the paint color is set to Red.

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

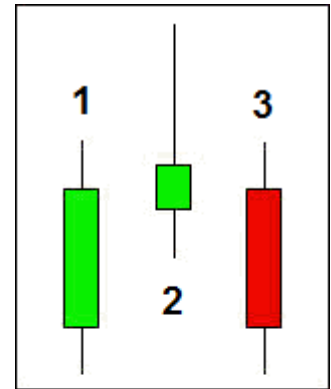
The paint color is set when the Show box is checked but no Marker is selected. The paint color is then used by a later row when its Color selection is Black.

3-Bar Signal

This example implements the following Sell Signal and Buy Signal.

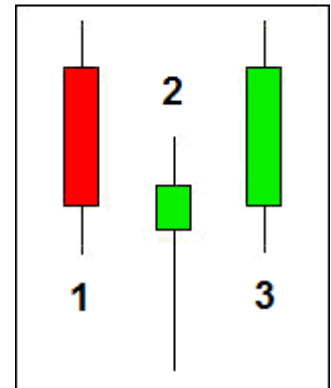
Rules for the Sell Signal:

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

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

Message Text: [Empty] | Font: [Black] | True: [Green] | False: [Red] | Panel: [Yellow]

Sound: Silent | Beep | Voice | WAV | Sound once per bar

Trading System: Price: [Empty] | 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	<input checked="" type="checkbox"/>	•	Pink
		AND	>	Prior Low	-1	<input checked="" type="checkbox"/>		
A	Expression	[Rule 1a] := (High[-1] > Prior High[-1]) AND (Low[-1] > Prior Low[-1])	<input checked="" type="checkbox"/>	•	Above High			
B	Expression	[Rule 1b] := (Prior High > High) AND (Prior Low > Low)	<input checked="" type="checkbox"/>	•	Above High			
C	Expression	[Rule 2] := Close[-2] > Open[-2]	<input checked="" type="checkbox"/>	•	Above High			
D	Expression	[Rule 3] := Close <= Open	<input type="checkbox"/>					
E	Flag	[0] := AND([Rule 1a], 4)	<input checked="" type="checkbox"/>		Above High			
F			<input type="checkbox"/>					
G	Expression	[Rule 1a] := (High[-1] < Prior High[-1]) AND (Low[-1] < Prior Low[-1])	<input checked="" type="checkbox"/>	•	Below Low			
H	Expression	[Rule 1b] := (Prior High < High) AND (Prior Low < Low)	<input checked="" type="checkbox"/>	•	Below Low			
I	Expression	[Rule 2] := Close[-2] < Open[-2]	<input checked="" type="checkbox"/>	•	Below Low			
J	Expression	[Rule 3] := Close >= Open	<input type="checkbox"/>					
K	Flag	[0] := AND([Rule 1a], 4)	<input checked="" type="checkbox"/>		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.

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.

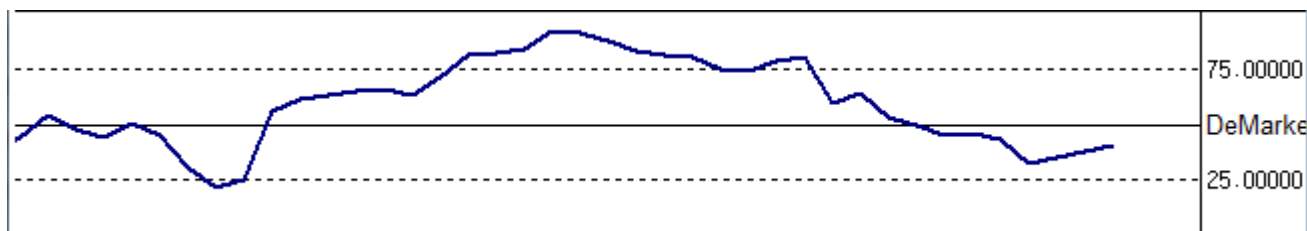
Formula

highdelta = IIF(High > PriorHigh, High - PriorHigh, 0);

lowdelta = IIF(Low < PriorLow, PriorLow - Low, 0);

DeMarker = 100 * Sum(highdelta, 13) / (Sum(lowdelta, 13) + Sum(highdelta, 13));

Study Name DeMarker	Study Location Sub Window	Marker Location Study Value	Label Location None	Grid Tab Default	<input type="checkbox"/> Use as Default <input type="checkbox"/> Draw Behind Bars <input type="checkbox"/> Privatize <input type="checkbox"/> Close Only <input type="checkbox"/> Show Values <input type="checkbox"/> Email <input type="checkbox"/> Auto Remove <input checked="" type="checkbox"/> Auto Completion
Study Scale Fixed Range	Scale High 100	Scale Low 0	Variables File -Demarker	Message Location None	
Message Text <input type="text"/>	Font Black	True Green	False Red	Panel White	
Sound <input checked="" type="radio"/> Silent <input type="radio"/> Beep <input type="radio"/> Voice <input type="radio"/> WAV <input type="checkbox"/> Sound once per bar	WAV file <input type="text"/> <input type="button" value="Browse"/>		Trading System Price: Last Quantity: 1 Commission: 0		
Category Action	Variable 22 HighSpread	Selection #1 & #3 = if ## then V := (#3 - #4) els	Op. [#] >	Selection #2 & #4 Prior High	
		(High			
A	Action	if (High > Prior High) then [HighSpread] := (High - Prior High) else		<input type="checkbox"/>	
B	Action	if (Prior Low > Low) then [LowSpread] := (Prior Low - Low) else		<input type="checkbox"/>	
C	Function	[SumHigh] := Sum([A], 13)		<input type="checkbox"/>	
D	Function	[SumLow] := Sum([B], 13)		<input type="checkbox"/>	
E	Expression	[Demarker] := (100 * [SumHigh]) / ([SumHigh] + [SumLow])		<input checked="" type="checkbox"/>	Study Value

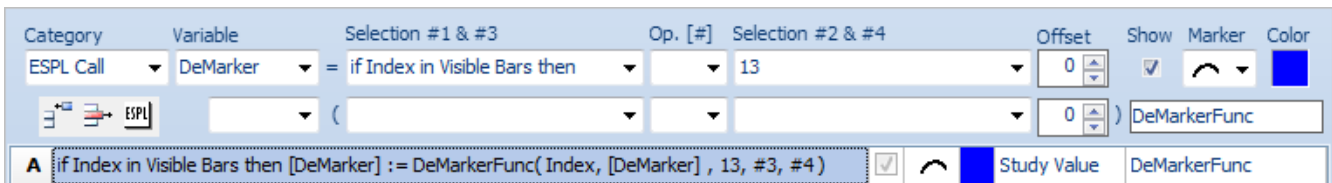


C – The Sum function references the values on row A, and not the variable HighDelta. The variable has a single value, whereas, row A has a series of values that can be summed.

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

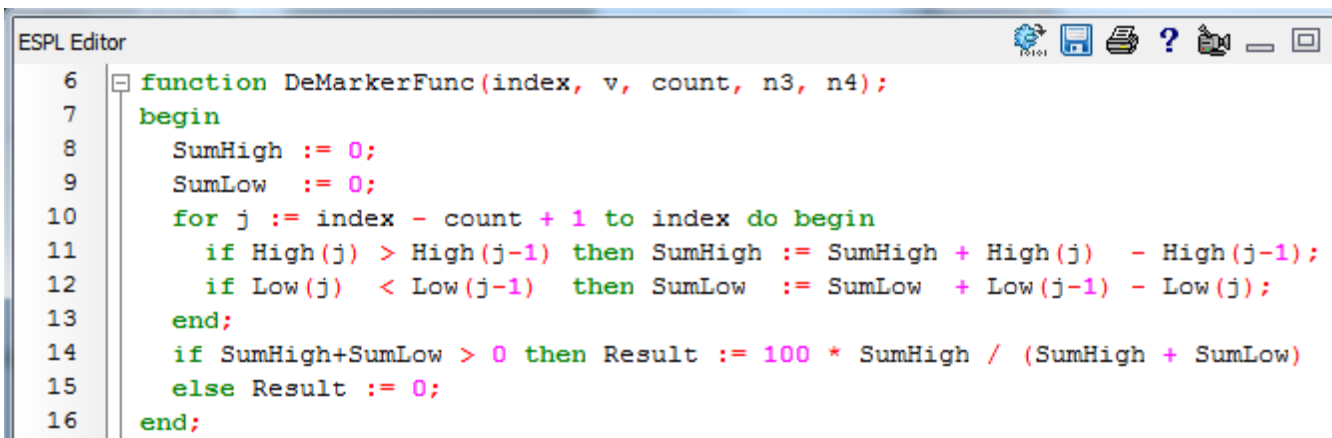
ESPL Script

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.



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 DYOpout(10,idx, High(idx) - High(idx-1));
21     if Low(idx) < Low(idx-1) then DYOpout(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:\Ensign10\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.

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.

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
Study	12 Ave>	= AVE 3,17 HH	d	1st line >= 2nd line	0	<input type="checkbox"/>		
A	Study	[Ave>] := AVE.1st line >= 2nd line				<input type="checkbox"/>		
B	Study	[SMI>] := SMI.SMI >= Average				<input type="checkbox"/>		
C	Study	[RSI>60] := (RSI.RSI value) > 60				<input type="checkbox"/>		
D	Study	[RSI<40] := (RSI.RSI value) < 40				<input type="checkbox"/>		
E	Action	if ([Ave>]) then marker size 2 in column 4				<input checked="" type="checkbox"/>		90% Grid AVE
F	Action	if (Not [Ave>]) then marker size 2 in column 4				<input checked="" type="checkbox"/>		90% Grid AVE
G	Action	if ([SMI>]) then marker size 2 in column 4				<input checked="" type="checkbox"/>		70% Grid SMI
H	Action	if (Not [SMI>]) then marker size 2 in column 4				<input checked="" type="checkbox"/>		70% Grid SMI
I	Action	if ([RSI>60]) then marker size 2 in column 4				<input checked="" type="checkbox"/>		50% Grid RSI
J	Action	if (Not [RSI<40]) then marker size 2 in column 4				<input checked="" type="checkbox"/>		50% Grid RSI
K	Action	if Not ([RSI>60] OR [RSI<40]) then marker size 2 in column 4				<input checked="" type="checkbox"/>		50% Grid RSI

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.

The screenshot displays the Strategy Tester configuration window. Key settings include:

- Study Name:** Study Consensus 2
- Study Location:** Chart
- Marker Location:** 25% Grid
- Label Location:** Column 7
- Grid Tab:** Default
- Study Scale:** Chart Scale
- Variables File:** -Consensus
- Message Location:** None
- Message Text:** (Empty field)
- Font:** True (Green), False (Red), Panel (Yellow)
- Sound:** Silent (selected)
- Trading System:** Last (Price), 1 (Quantity), 0 (Commission)

The Strategy Tester table below shows the logic for the Buy and Sell signals:

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
Action	0	= if ## then marker size Offse		4	2	<input checked="" type="checkbox"/>		BUY
		([BuyFlag]			0	<input checked="" type="checkbox"/>		SELL
A	Expression	[BuyFlag] := ([Ave>] AND [SMI>]) AND [RSI>60]				<input type="checkbox"/>		
B	Expression	[SellFlag] := [RSI<40] AND Not ([Ave>] OR [SMI>])				<input type="checkbox"/>		
C	Action	if ([BuyFlag]) then marker size 2 in column 4				<input checked="" type="checkbox"/>		25% Grid BUY
D	Action	if ([SellFlag]) then marker size 2 in column 4				<input checked="" type="checkbox"/>		25% Grid SELL
E	Action	if Not ([BuyFlag] OR [SellFlag]) then marker size 2 in column 4				<input checked="" type="checkbox"/>		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.

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
A Study	13 Flag	= AVE 2.9 HH g		1st line X<> 2nd line	0	<input checked="" type="checkbox"/>	ZON	Yellow
B Action		if [13 Flag] then Play Sound				<input type="checkbox"/>		

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'.

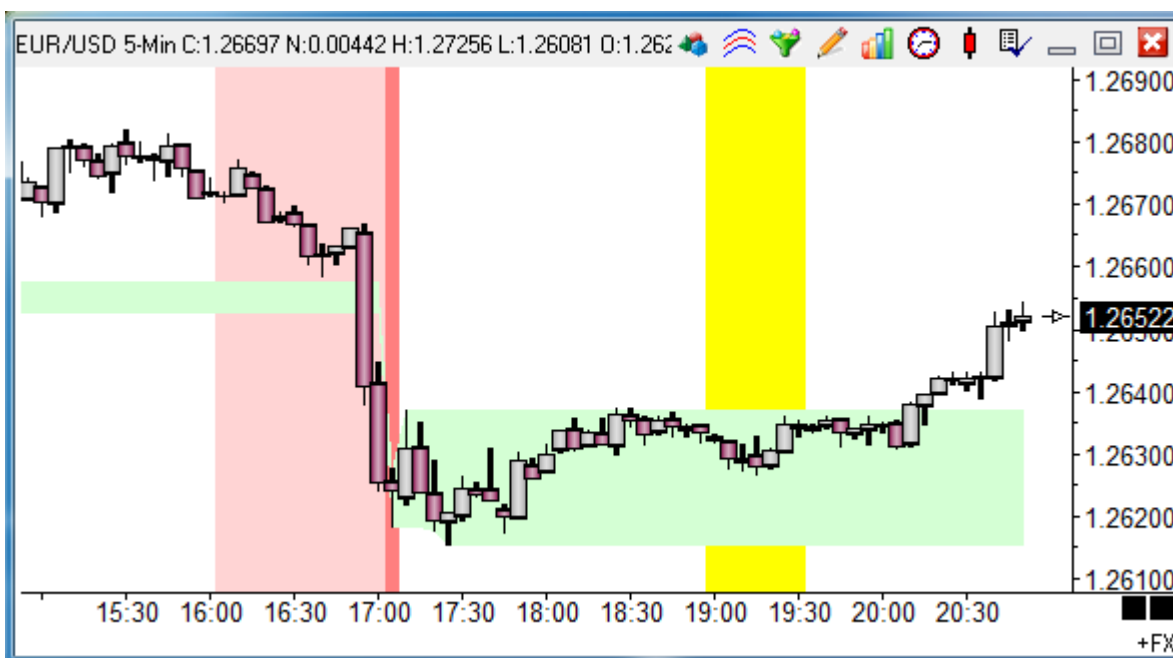
B – When the flag from Row A is True, the DYO will play the sound in the Sound frame. The example shows the Voice bullet is selected, which will read the text from the Study Name field. The words 'Study Crossing Alert' will be spoken.

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.

Price and Time Bands

This example demonstrates coloring a price range and time periods.



Study Name: Price and Time Bands

Study Location: Chart

Marker Location: Study Value

Label Location: None

Grid Tab: Default

Study Scale: Chart Scale

Variables File: Ensign

Message Location: None

Message Text:

Font: True (black), False (green), Panel (red), (yellow)

Sound: Silent Beep WAV Copy to Clipboard Sound once per bar

WAV file:

Trading System: Price: Last, Quantity: 0, Commission: 0

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show Marker	Color
Flag	0	= #2 between #3 & #4		Time	0	<input checked="" type="checkbox"/>	ZON
		(1900		1930	0		
A	Chart Value	[0] := 30 Minute High after 2nd Open	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	 Study Value
B	Chart Value	[0] := 30 Minute Low after 2nd Open	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
C			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
D	Flag	[0] := 1st bar of New Trade Day	<input checked="" type="checkbox"/>	ZON	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	 Study Value
E	Flag	[0] := In 60 min before Session 1 End	<input checked="" type="checkbox"/>	ZON	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	 Study Value
F	Flag	[0] := Time between 1900 & 1930	<input checked="" type="checkbox"/>	ZON	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	 Study Value

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

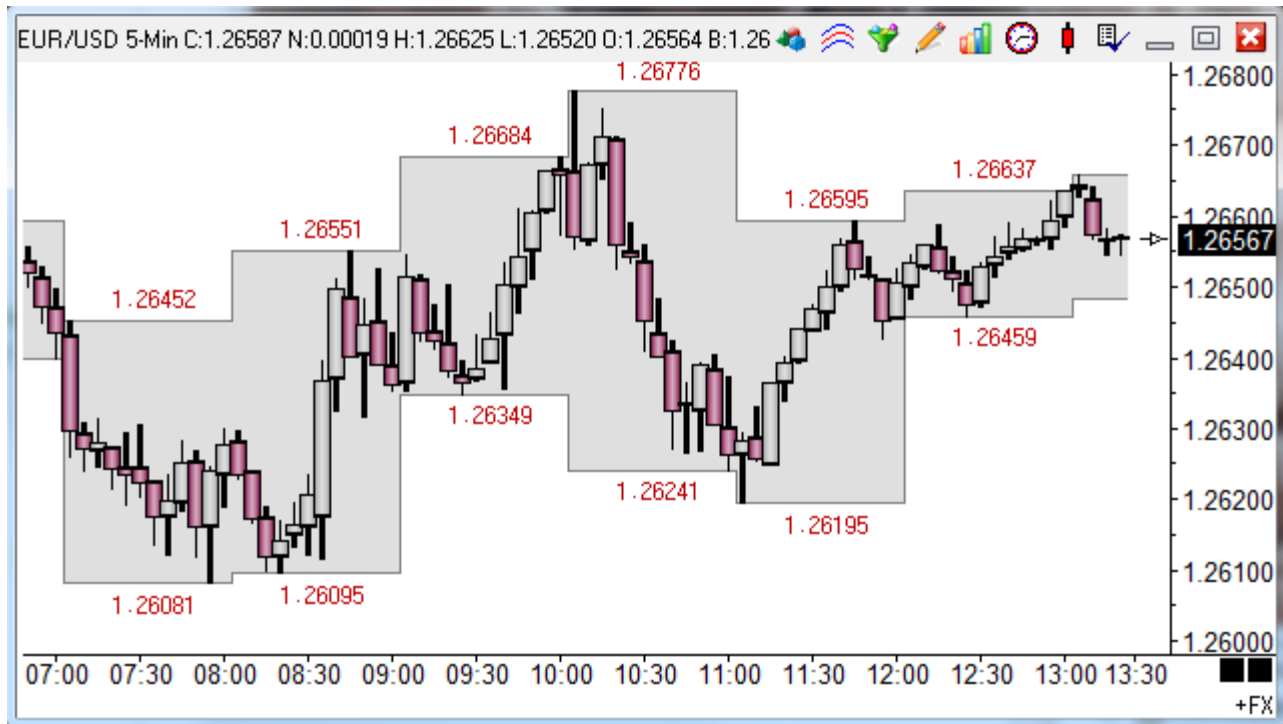
The Show box must to be checked for a row's Marker and/or Label to be shown.

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.

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 Hourly Bars	Study Location Chart	Marker Location Study Value	Label Location None	Grid Tab Default	<input type="checkbox"/> Use as Default
Study Scale Chart Scale			Variables File -HourlyBars	Message Location None	<input checked="" type="checkbox"/> Draw Behind Bars
Message Text <input type="text"/>	Font [Color Swatches]	True [Color Swatches]	False [Color Swatches]	Panel [Color Swatches]	<input type="checkbox"/> Privatize
Sound <input checked="" type="radio"/> Silent <input type="radio"/> Beep <input type="radio"/> Voice <input type="radio"/> WAV <input type="checkbox"/> Sound once per bar	WAV file <input type="text"/> <input type="button" value="Browse"/>	Trading System Price Last		Quantity 1	Commission 0
Category Chart Value	Variable 61 High	Selection #1 & #3 = #2 Minute High	Op. [#] =	Selection #2 & #4 60	Offset 0
					Show <input checked="" type="checkbox"/>
					Marker [Marker Icon]
					Color [Color Swatch]

	Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
A	Chart Value	[High] := 60 Minute High					<input checked="" type="checkbox"/>	[Marker Icon]	Study Value
B	Chart Value	[Low] := 60 Minute Low					<input checked="" type="checkbox"/>	[Marker Icon]	Study Value
C	Expression	[0] := [High]					<input checked="" type="checkbox"/>	[Marker Icon]	Study Value
D	Expression	[Minutes] := (Minutes from Midnight / 60) - 0.55					<input type="checkbox"/>		
E	Function	[0] := trunc([Minutes])					<input type="checkbox"/>		
F	Flag	[TurnsUp] := [E] turns up					<input type="checkbox"/>		
G	Branching	if Not [TurnsUp] then abort execution					<input type="checkbox"/>		
H	Expression	[0] := [High] + 0.0003					<input checked="" type="checkbox"/>	LBL	Study Value [A]
I	Expression	[0] := [Low] - 0.0003					<input checked="" type="checkbox"/>	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.

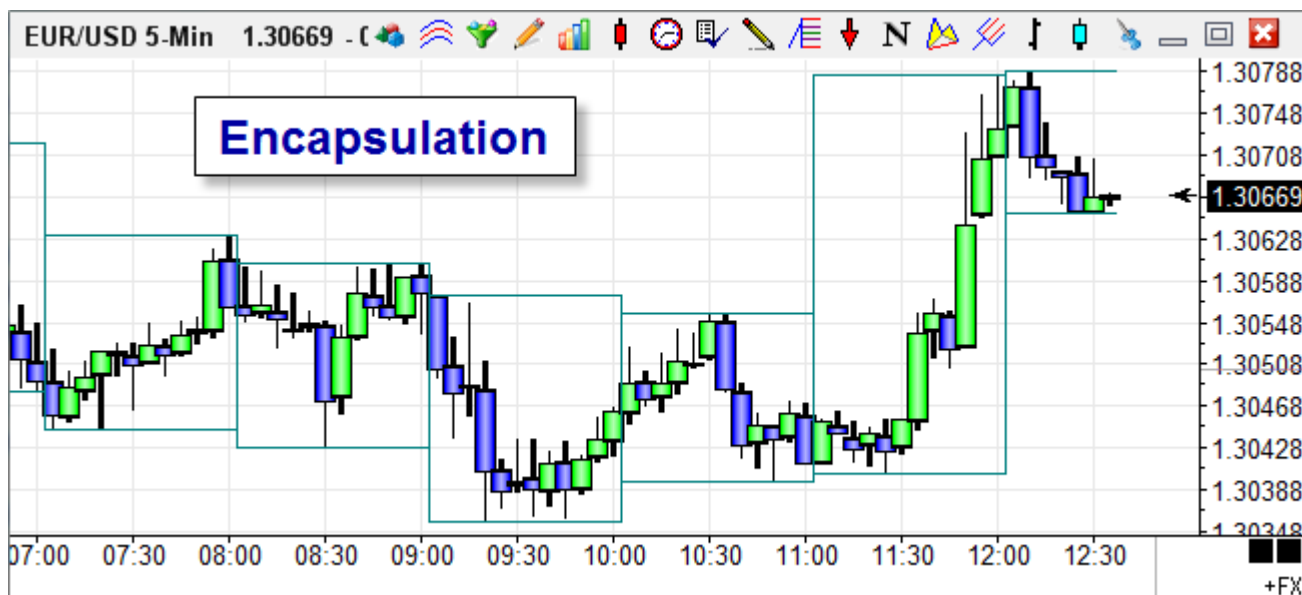
Row D, E, F and G accomplish limiting the values to show once per 60 min period instead of on every bar.

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.

Encapsulation

An alternate way to encapsulate the bars is to use the Flute marker on the first row of a 2 row pair that find the Highs and Lows. See the next example.



Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
Chart Value	61 High	= #2 Minute High		60	0	<input checked="" type="checkbox"/>		
<div style="display: flex; justify-content: space-between; align-items: center;"> </div>								
A	Chart Value	[61 High] := 60 Minute High				<input checked="" type="checkbox"/>		Study Value
B	Chart Value	[60 Low] := 60 Minute Low				<input type="checkbox"/>		

A - B – The high and low for a 60-minute period are returned. The Flute marker will draw horizontal lines at the price of this row and the price of the following row. Vertical lines will be drawn whenever either of these prices change. The resulting visual effect is a frame that encapsulates the time period.

The marker uses 2 values and these values come from the row with the marker and the next DYO row. Have the High be on the first row, and the Low on the next row, as shown in the example.

Note: While the last encapsulation frame is building, new highs or lows can extend the range. This will trigger excess vertical lines to be drawn. Press the CTRL-ALT-R keys to force the DYO to recalculate, which will reset all bars in the last encapsulation to the new range.

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.



Study Name Pesavento Break	Study Location Chart	Marker Location Study Value	Label Location None	Grid Tab Default	<input type="checkbox"/> Use as Default <input type="checkbox"/> Draw Behind Bars <input type="checkbox"/> Privatize <input type="checkbox"/> Close Only <input type="checkbox"/> Show Values <input type="checkbox"/> Email <input type="checkbox"/> Auto Remove		
Study Scale Chart Scale	Variables File Ensign	Message Location None					
Message Text <input type="text"/>	Font True False Panel	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
Sound <input checked="" type="radio"/> Silent <input type="radio"/> Beep <input type="radio"/> WAV <input type="radio"/> Copy to Clipboard <input type="checkbox"/> Sound once per bar WAV file <input type="text"/> <input type="button" value="Browse"/>	Trading System Price Last		Quantity 0	Commission 0			
Category Expression	Variable 0	Selection #1 & #3 = High	Op. [#] >	Selection #2 & #4 [Swing]	Offset 0	Show Marker <input checked="" type="checkbox"/> HLT	Color [Pink]
<input type="button" value="AND"/> AND ([Rising]					0		
A	Study	[Swing] := Pesavento.Swing 3 value	<input checked="" type="checkbox"/>	—	Study Value		
B	Study	[Rising] := Pesavento.Trend line rising	<input type="checkbox"/>				
C	Expression	[0] := (High > [Swing]) AND [Rising]	<input checked="" type="checkbox"/>	HLT	Study Value		
D	Expression	[0] := (Low < [Swing]) AND Not [Rising]	<input checked="" type="checkbox"/>	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.

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]

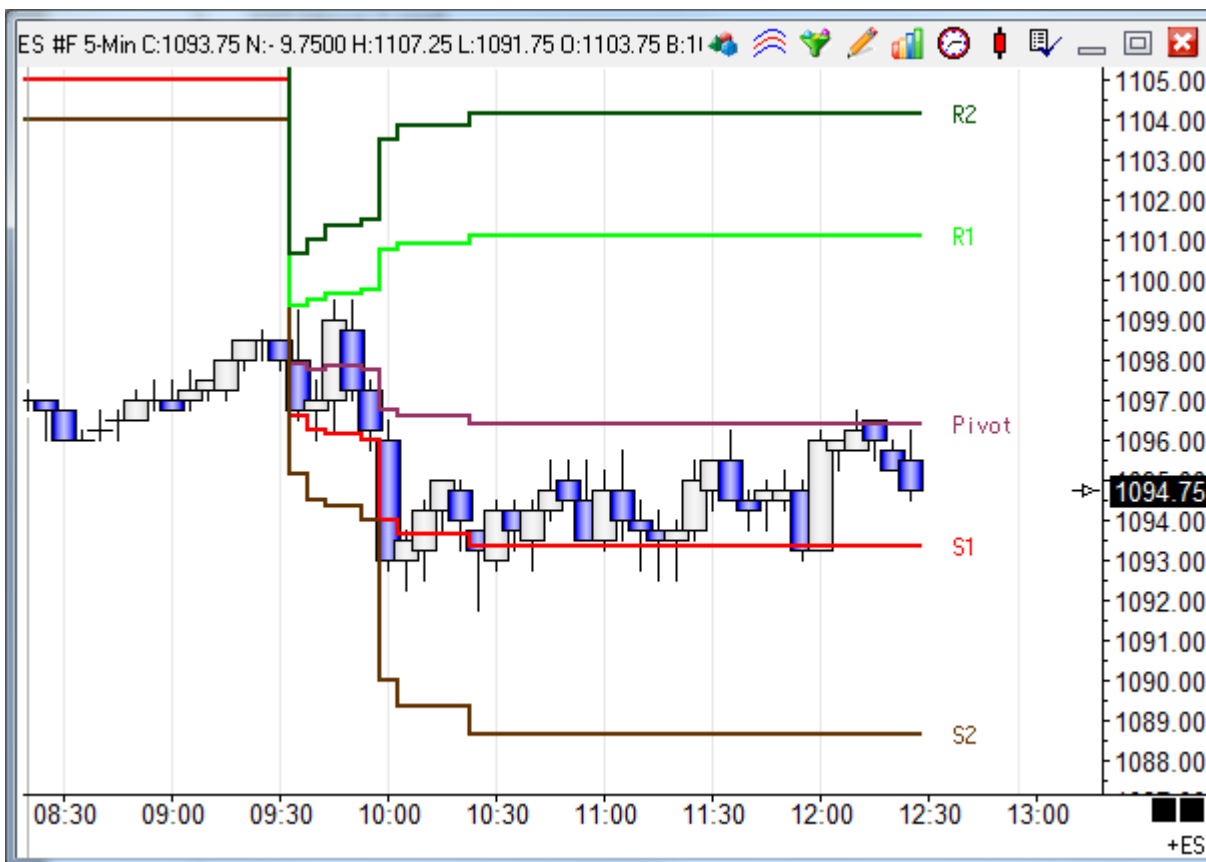
L [1st Hour Low]

1S [1st Support] = Pivot - (High-Pivot)

2S [2nd Support] = Pivot - Range

3S [3rd Support] = 1S – Range

4S [4th Support] = 2S – Range



Study Name Dynamic S & R Lines	Study Location Chart	Marker Location Study Value	Label Location Column 2	Grid Tab Default	<input type="checkbox"/> Use as Default <input type="checkbox"/> Draw Behind Bars <input type="checkbox"/> Privatize <input type="checkbox"/> Close Only <input type="checkbox"/> Show Values <input type="checkbox"/> Email <input type="checkbox"/> Auto Remove	
Study Scale Chart Scale	Variables File Ensign		Message Location None			
Message Text <input type="text"/>	Font	True	False	Panel		
Sound <input checked="" type="radio"/> Silent <input type="radio"/> Beep <input type="radio"/> WAV <input type="radio"/> Copy to Clipboard <input type="checkbox"/> Sound once per bar WAV file <input type="text"/> <input type="button" value="Browse"/>	Trading System Price Last		Quantity 0	Commission 0		
Category Chart Value	Variable 58 High60	Selection #1 & #3 = #2 Minute High after 1st Open	Op. [#] =	Selection #2 & #4 60	Offset 0	Show Marker <input type="checkbox"/>
<input type="button" value="EPL"/> (<input type="text"/>) <input type="button" value="EPL"/>						
A	Chart Value	[High60] := 60 Minute High after 1st Open	<input type="checkbox"/>			
B	Chart Value	[Low60] := 60 Minute Low after 1st Open	<input type="checkbox"/>			
C	Expression	[Open1] := Session 1 Open	<input type="checkbox"/>			
D	Function	[Pivot] := Simple Average([Open1], 3)	<input checked="" type="checkbox"/>		Study Value	Pivot
E	Expression	[S1] := (2 * [Pivot]) - [High60]	<input checked="" type="checkbox"/>		Study Value	S1
F	Expression	[S2] := [Pivot] - ([High60] - [Low60])	<input checked="" type="checkbox"/>		Study Value	S2
G	Expression	[R1] := (2 * [Pivot]) - [Low60]	<input checked="" type="checkbox"/>		Study Value	R1
H	Expression	[R2] := [Pivot] + ([High60] - [Low60])	<input checked="" type="checkbox"/>		Study Value	R2

A – Save the high of the 1st hour following the 1st session open in a variable named High60. 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 Low60.

C – Save the open of the 1st session in a variable named Open1.

D – Calculate the Pivot by averaging [Open1], [High60], and [Low60]. The 3 variables used are adjacent to each other with Open1 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.

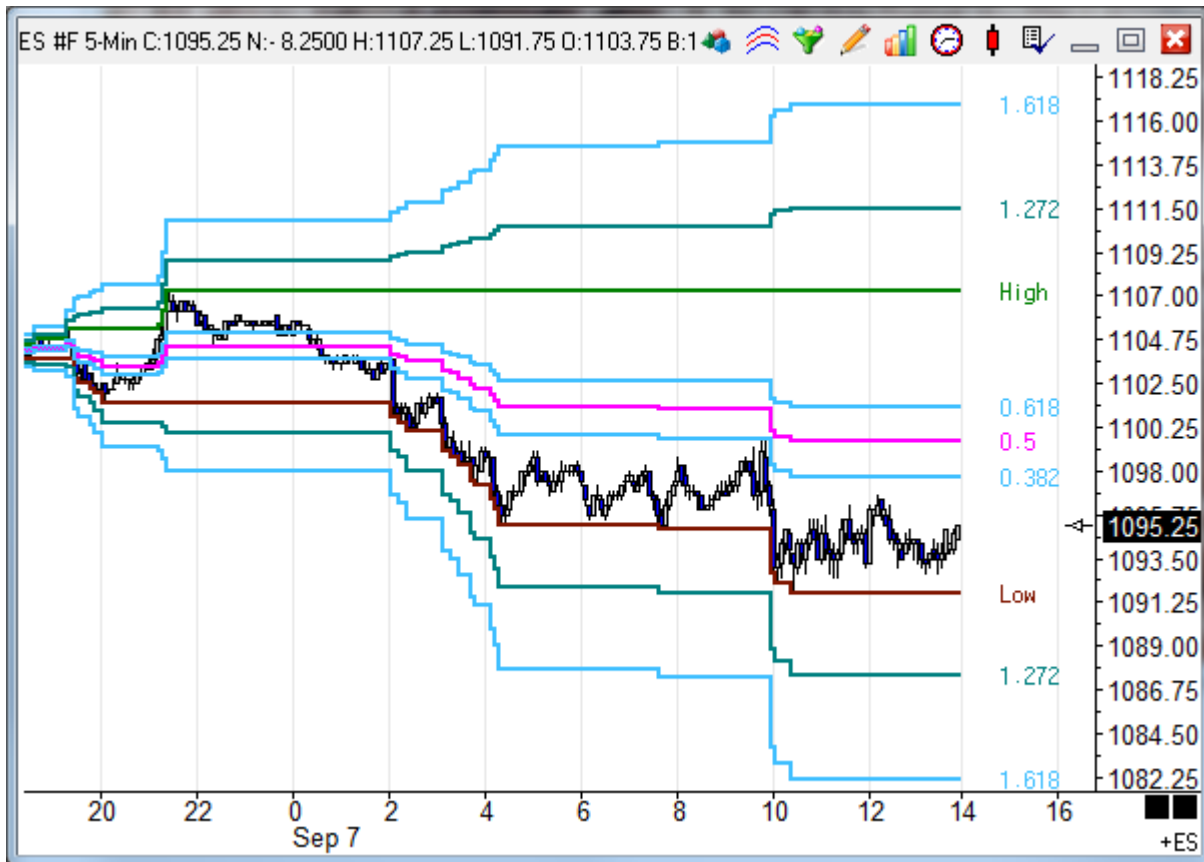
G – Calculate 1st Resistance.

H – Calculate 2nd Resistance.

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

Dynamic Fibonacci Levels

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



Study Name: Dynamic Fibonacci Levels | Study Location: Chart | Marker Location: Study Value | Label Location: Column 2 | Grid Tab: Default

Study Scale: Chart Scale | Variables File: Ensign | Message Location: None

Message Text: [] | Font: True (Black), False (Green), Panel (Red, Yellow)

Sound: Silent | Beep | WAV | Copy to Clipboard | Sound once per bar

Trading System: Price: Last | Quantity: 0 | Commission: 0

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
Function	0	=	Fib(#2 * (#3 - #4) + #4)	1.618	0	<input checked="" type="checkbox"/>		Blue
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>								
		(Today High)	Today Low	0		1.618
A	Expression	[0] := Today High	<input checked="" type="checkbox"/>		Study Value	<input checked="" type="checkbox"/>		High
B	Expression	[0] := Today Low	<input checked="" type="checkbox"/>		Study Value	<input checked="" type="checkbox"/>		Low
C	Function	[0] := Fib(1.618 * (Today High - Today Low) + Today Low)	<input checked="" type="checkbox"/>		Study Value	<input checked="" type="checkbox"/>		1.618
D	Function	[0] := Fib(1.272 * (Today High - Today Low) + Today Low)	<input checked="" type="checkbox"/>		Study Value	<input checked="" type="checkbox"/>		1.272
E	Function	[0] := Fib(0.618 * (Today High - Today Low) + Today Low)	<input checked="" type="checkbox"/>		Study Value	<input checked="" type="checkbox"/>		0.618
F	Function	[0] := Fib(0.5 * (Today High - Today Low) + Today Low)	<input checked="" type="checkbox"/>		Study Value	<input checked="" type="checkbox"/>		0.5
G	Function	[0] := Fib(0.382 * (Today High - Today Low) + Today Low)	<input checked="" type="checkbox"/>		Study Value	<input checked="" type="checkbox"/>		0.382
H	Function	[0] := Fib(1.272 * (Today Low - Today High) + Today High)	<input checked="" type="checkbox"/>		Study Value	<input checked="" type="checkbox"/>		1.272
I	Function	[0] := Fib(1.618 * (Today Low - Today High) + Today High)	<input checked="" type="checkbox"/>		Study Value	<input checked="" type="checkbox"/>		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.

Section Font Size

The screenshot shows the 'Design Your Own Study' window. At the top, a chart displays three sections labeled 'Hello' in different font sizes and styles. Below the chart, the 'Design Your Own Study' dialog box is open, showing various configuration options. The 'Message Text' field is empty. The 'Sound' section has 'Silent' selected. The 'Trading System' section has 'Last' selected. The 'Category' section shows a table with three rows (A, B, C) defining actions for different sections.

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show Marker	Color
Action	0	= Section Font Size := #2		10	0	<input checked="" type="checkbox"/>	
		{			0		1 Hello
A	Action	Section Font Size := 10				<input checked="" type="checkbox"/>	Study Value 1 Hello
B	Action	Section Font Size := 12				<input checked="" type="checkbox"/>	Study Value 2 Hello
C	Action	Section Font Size := -14				<input checked="" type="checkbox"/>	Study Value 3 Hello

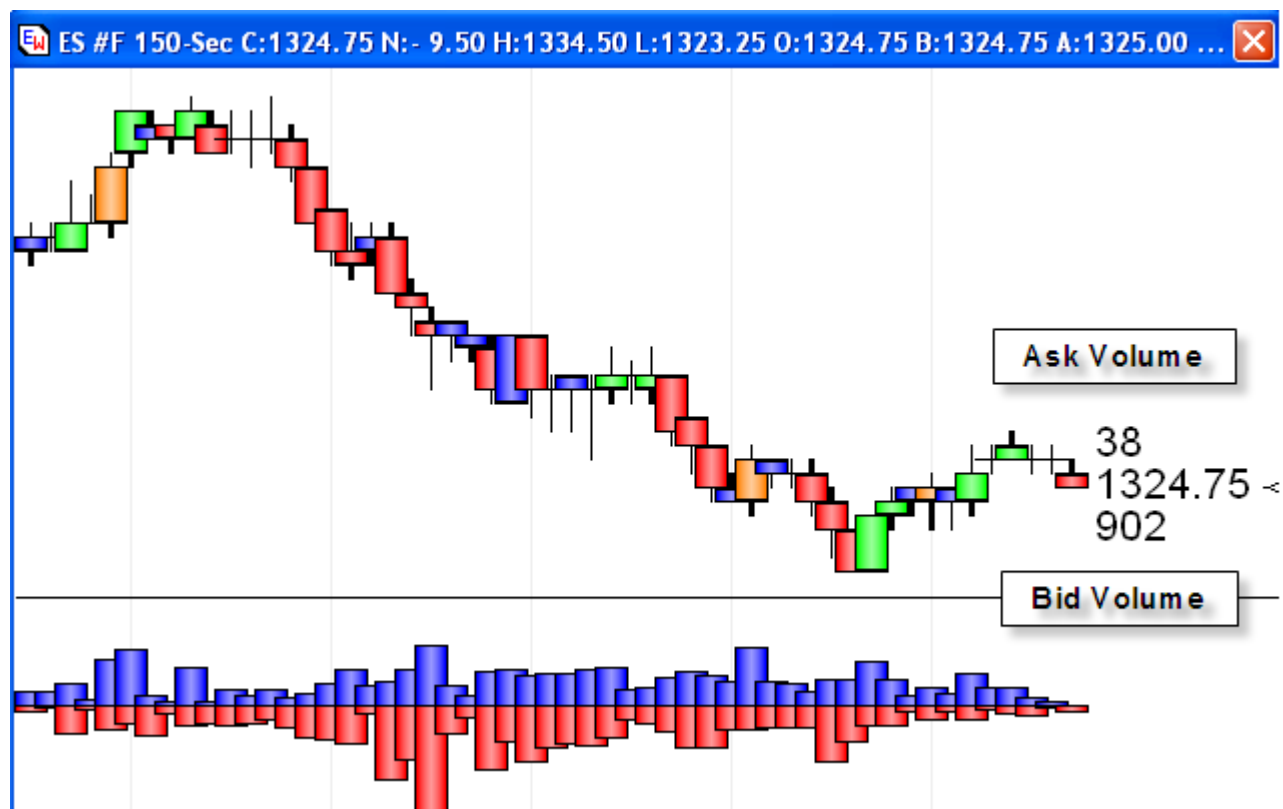
A – Set the font size to 10 for section |1| and put the message Hello in that section. The |1| on row A is a tag to use section |1|. The tag is the vertical line character on both sides of a single digit in the range 1..7. The example has the 'Hello' text in the row A label, but the text is optional, and could have been on some other row or in the Message Text edit box.

B – Set the section |2| font size to 12.

C – Set the section |3| font size to Bold 14. Note the use of the negative sign to create Bold.

Long Messages - A long message will start at the left edge of the section and overflow a section in the rightward direction. This means it affects its right side neighbor but not the left side neighbor. Use spaces to pad the left or right side of a message to shift its centering.

Label Font Size

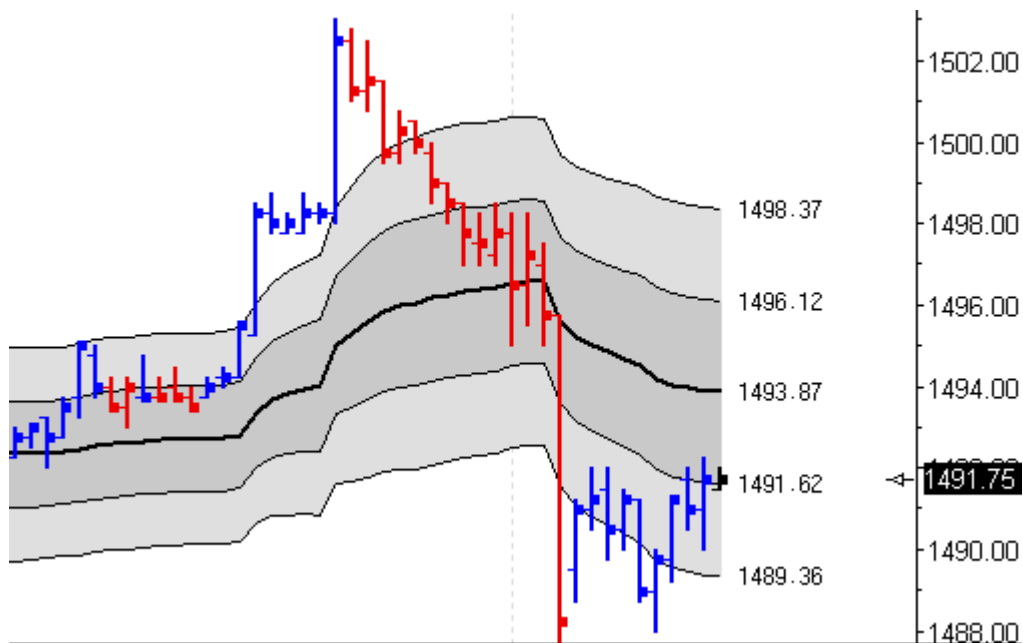


Study Name Design Your Own	Study Location Chart	Marker Location Last	Label Location Column 2	Grid Tab Default	<input type="checkbox"/> Use as Default <input type="checkbox"/> Draw Behind Bars <input type="checkbox"/> Privatize <input type="checkbox"/> Close Only <input type="checkbox"/> Show Values		
Study Scale Chart Scale	Message Location None		Message Text <input type="text"/>				
Sound <input type="radio"/> Silent <input type="radio"/> Beep <input type="radio"/> WAV <input type="radio"/> Copy to Clipboard <input type="checkbox"/> Sound once per bar WAV file <input type="text"/> <input type="button" value="Browse"/>		Font True False Panel <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>		Auto Remove <input type="checkbox"/>	Email <input type="checkbox"/>		
Trading System Price Last			Quantity 0	Commission 0			
Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show Marker	Color
Action	0	= Label Font Size := #2		16	0	<input checked="" type="checkbox"/>	█
[SPL]		(0		[\$.].[\$C].[\$B.]
A Action	Label Font Size := 16		<input checked="" type="checkbox"/>	█	Last		[\$.].[\$C].[\$B.]

A – Shows a label in a size 16 font. The label has 3 lines showing the Ask Volume, current price, and Bid Volume. The [\$.] tag references the Ask Volume and formats the value as an integer. The commas split the label into separate lines. The vertical placement is the Marker Location of the Last price. The horizontal placement is the Label Location of Column 2.

[read more >> Reference Tags](#)

Study Labels



Study Name: Study Labels | Study Location: Chart | Marker Location: Study Value | Label Location: Column 2 | Grid Tab: Default

Study Scale: Chart Scale | Message Location: None

Message Text: | Font: True (Black), False (Green), Panel (Red), Auto Remove: | Email:

Sound: Silent Beep WAV Copy to Clipboard Sound once per bar | WAV file: Browse

Trading System: Price: Last | Quantity: 0 | Commission: 0

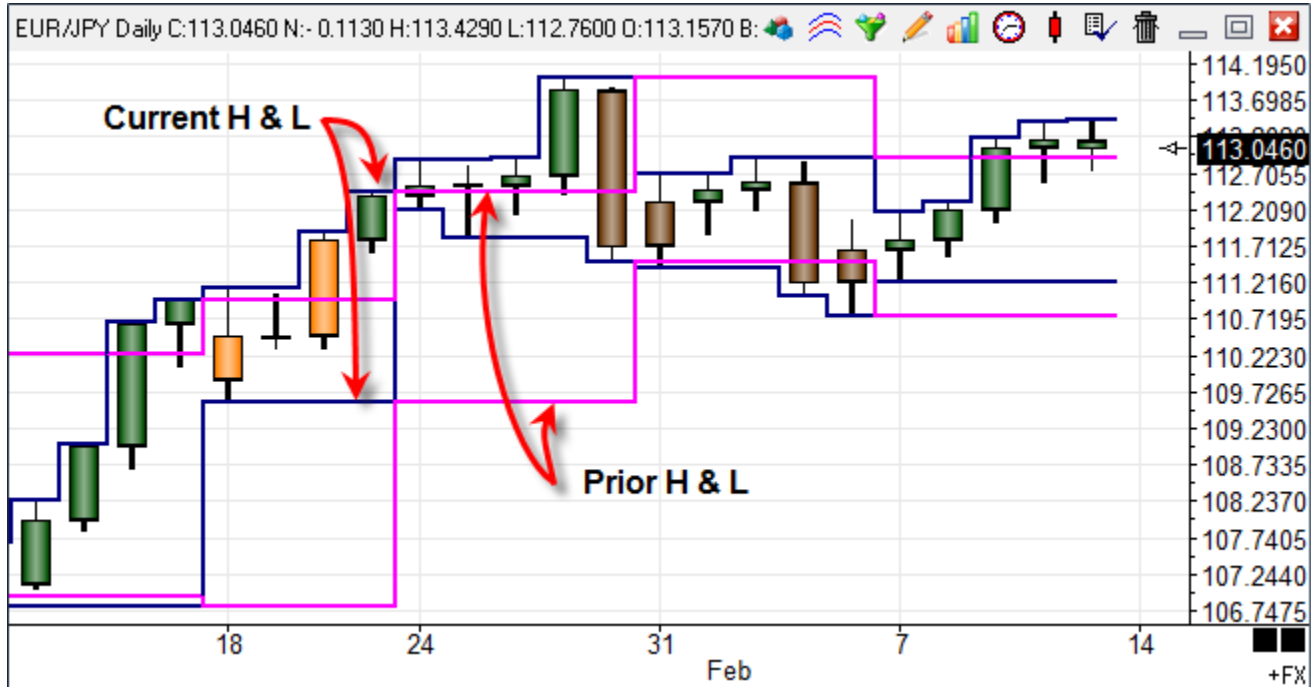
Category	Variable	Selection #1 & #3	Op. (#)	Selection #2 & #4	Offset	Show	Marker	Color
Study	0	= VWAP 1,2	f	2nd Upper band	0	<input checked="" type="checkbox"/>	<input type="text"/>	Black
<input type="text"/> (<input type="text"/>) [A]								
A	Study	[0] := VWAP.2nd Upper band	<input checked="" type="checkbox"/>	Study Value	[A]			
B	Study	[0] := VWAP.1st Upper band	<input checked="" type="checkbox"/>	Study Value	[B]			
C	Study	[0] := VWAP.Center line	<input checked="" type="checkbox"/>	Study Value	[C]			
D	Study	[0] := VWAP.1st Lower band	<input checked="" type="checkbox"/>	Study Value	[D]			
E	Study	[0] := VWAP.2nd Lower band	<input checked="" type="checkbox"/>	Study Value	[E]			

A - E – Show a study value as a label in the Column 2 Label Location. The vertical placement is the Study Value Marker Location. Each label is using the black color in this example. The VWAP study is ahead of the DYO on the chart's object list so the DYO can reference the VWAP study.

The label font size could have been changed using Action | Label Font Size := #2.

Weekly High and Low

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



Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Function	233 CurrentWee	= Highest(#2, [#])	229	High	0
Flag	[NewWeek] := 1st bar of New Week				
Action	if ([NewWeek]) then [PriorWeekHigh] := [CurrentWeekHigh]				
Action	if ([NewWeek]) then [PriorWeekLow] := [CurrentWeekLow]				
Action	if ([NewWeek]) then [BarCount] := 0				
Expression	[BarCount] := [BarCount] + 1				
Function	[CurrentWeekHigh] := Highest(High, [BarCount])				Study Value
Function	[CurrentWeekLow] := Lowest(Low, [BarCount])				Study Value
Expression	[0] := [PriorWeekHigh]				Study Value
Expression	[0] := [PriorWeekLow]				Study Value

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]))	<input checked="" type="checkbox"/>		Study Value
G	Function	[CurrentWeekLow] := Minimum(Low, ([CurrentWeekLow]))	<input checked="" type="checkbox"/>		Study Value
H	Expression	[0] := [PriorWeekHigh]	<input checked="" type="checkbox"/>		Study Value
I	Expression	[0] := [PriorWeekLow]	<input checked="" type="checkbox"/>		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.

Draw Statements

A DYO can be used to draw lines, circles, ellipses, arcs, and rectangles on a chart to represent trends, support and resistance levels, and target zones.

Some draw line effects can be accomplished using the horizontal line markers found on the Marker drop down list. There are horizontal markers with a short length, long length, and some that extend left or right to the edge of the chart.

A DYO also offers this set of additional draw statements:

These statements conditionally draw when the expression evaluates to True. The Show check box on the row must also be checked.

```
if #2 oper 0 then MoveTo( #3, #4 )
if #2 oper 0 then LineTo( #3, #4 )
if #2 oper 0 then HorzLine( #3, #4 )
if #2 oper 0 then VertLine( #3, #4 )
if #2 oper 0 then Rectangle( #3, #4 )
if #2 oper 0 then Circle( #3, #4 )
if #2 oper 0 then Ellipse( #3, #4 )
if #2 oper 0 then Arc( #3, #4 )
```

Data Point

A data point consists of a horizontal location based on a bar's index, and a vertical location based on a price. Both pieces of information must be provided.

The horizontal index is entered in the Selection #3 field. The price is entered in Selection #4.

If the #3 value is less than 11, then it is an offset from the current bar's index. If the value is greater than or equal to 11, then it is the bar index for the data point. The index for the current bar can be read using the Bar Index selection in the field selection list.

MoveTo

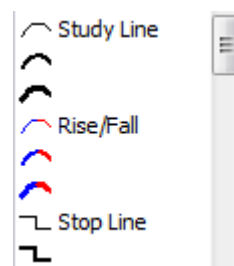
This statement moves the pen to a data point without drawing a line. MoveTo is used as a companion statement with the LineTo, Rectangle, Ellipse, and Arc statements that use two data points for their construction.

The MoveTo statement can also be used to draw a marker at its data point.

LineTo

This statement draws a line from the MoveTo data point to the LineTo data point. The pen location is left at the LineTo data point. The line's color is the color set on the DYO row. The line thickness is based on the marker thickness. One of the first 8 line drawing markers must be selected.

The curve markers will draw a straight line between the 2 points. The stair step markers will draw the line between the 2 points using a stair step.



HorzLine

HorzLine will draw a horizontal line at the #4 price. The line will begin with current bar and extend rightward. The #3 field is used to specify the length of the line. If the #3 value is positive the length of the line is in bar spacing units. If the #3 value is negative, the value is the length of the line in pixels.

VertLine

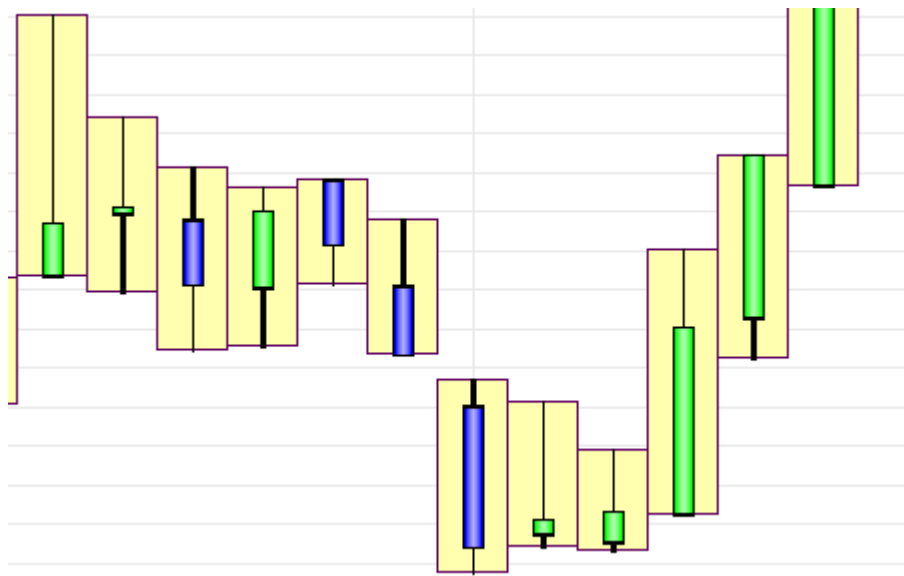
VertLine draws a vertical line between 2 prices. The #3 value is the 1st price, and the #4 value is the 2nd price. The vertical line is aligned with the current bar.

Rectangle

The Rectangle statement draws a frame between the MoveTo data point and the Rectangle data point. The thickness of the frame is based on the marker thickness, the same as discussed for the LineTo statement.

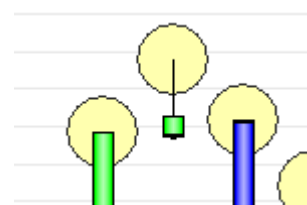
If the marker selected is one of the Rise/Fall curves, then the rectangle will be filled with the color selected on the next DY0 row. This example draws a filled rectangle around each bar. Note the index offset for the MoveTo is a half bar space leftward (-0.5), and the offset for the Rectangle is a half bar space rightward (0.5).

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
Action	0	= if #2 oper 0 then Rectangle		True	0	<input checked="" type="checkbox"/>		
		{ 0.5		High	0			
A	Action	if True then MoveTo(-0.5, Low)	<input checked="" type="checkbox"/>					Paint Color
B	Action	if True then Rectangle(0.5, High)	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>		Study Value
C	Expression		<input type="checkbox"/>					



Circle

The Circle statement is similar to the HorzLine statement. A circle is drawn aligned with the current bar, at the #4 price. The radius of the circle is the #3 value. A positive value for #3 is a radius in bar spacing units. A negative value for #3 is a radius in pixels.



If the marker selected is one of the Rise/Fall curves, then the circle will be filled with the color selected on the next DYO row.

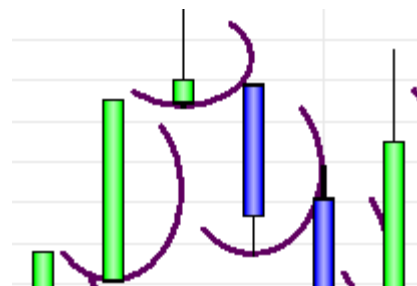
Ellipse

This statement is similar to the discussion for Rectangle. An ellipse is drawn bounded by the rectangle specified by the MoveTo data point and the Ellipse data point. The thickness of the perimeter is based on the marker selected. The ellipse will be filled if a Rise/Fall marker is used. The color for the fill is from the next DYO row.



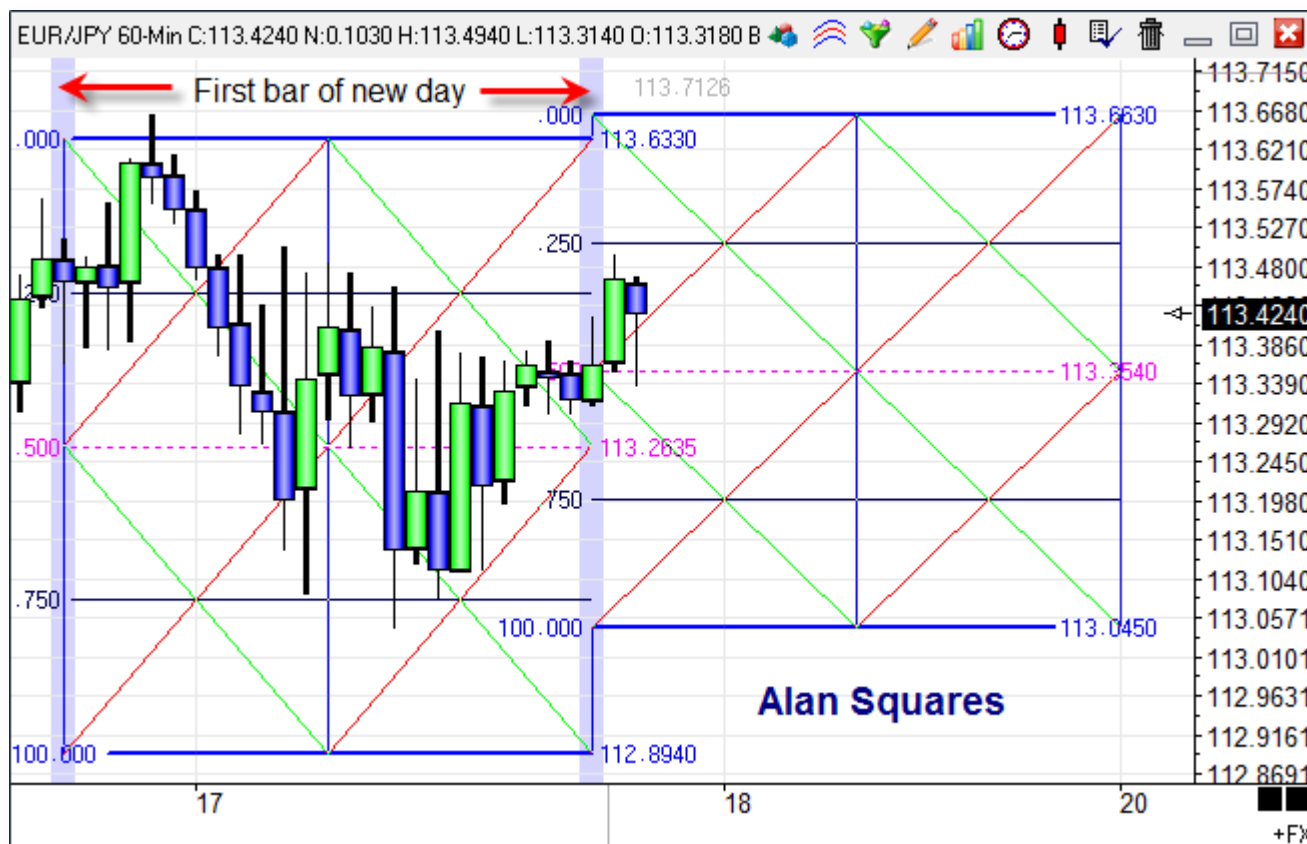
Arc

Arc is a variation of the ellipse. A section of the ellipse perimeter is drawn. If the Arc price is higher than the MoveTo price, then a bottom arc is drawn. If the Arc price is below the MoveTo price, then a top arc is drawn.



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

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 this DYO's execution when line A is false.

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 from 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		<input checked="" type="checkbox"/> ZON <input type="checkbox"/> Stuc
B	Action	if (Not [A]) then [ResetFlag] := False		<input type="checkbox"/>
C	Branching	if Not [A] then abort execution		<input type="checkbox"/>
D	Branching	if [ResetFlag] then abort execution		<input type="checkbox"/>
E	Study	[Handle] := Alan.Handle		<input type="checkbox"/>
F	Action	object([Handle]).Set Point A([PriorIndex], [PriorYH])		<input type="checkbox"/>
G	Action	object([Handle]).Set Point B(Bar Index, [PriorYL])		<input type="checkbox"/>
H	Expression	[ResetFlag] := True		<input type="checkbox"/>
I	Expression	[PriorIndex] := Bar Index		<input type="checkbox"/>
J	Expression	[PriorYH] := Yesterday High		<input type="checkbox"/>
K	Expression	[PriorYL] := Yesterday Low		<input type="checkbox"/>

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 this DYO's 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.

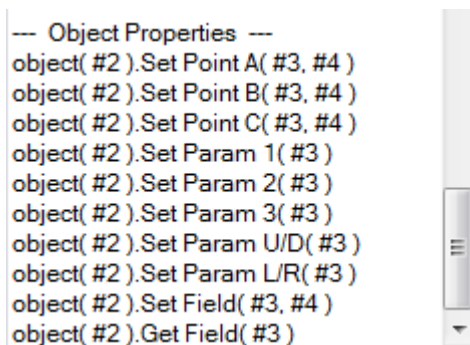
H – Set the flag on the pass through the DYO that moves the Alan Square and updates the PriorYH and PriorYL values. By setting the [ResetFlag] to True, subsequent calls will be blocked.

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 Properties --
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.

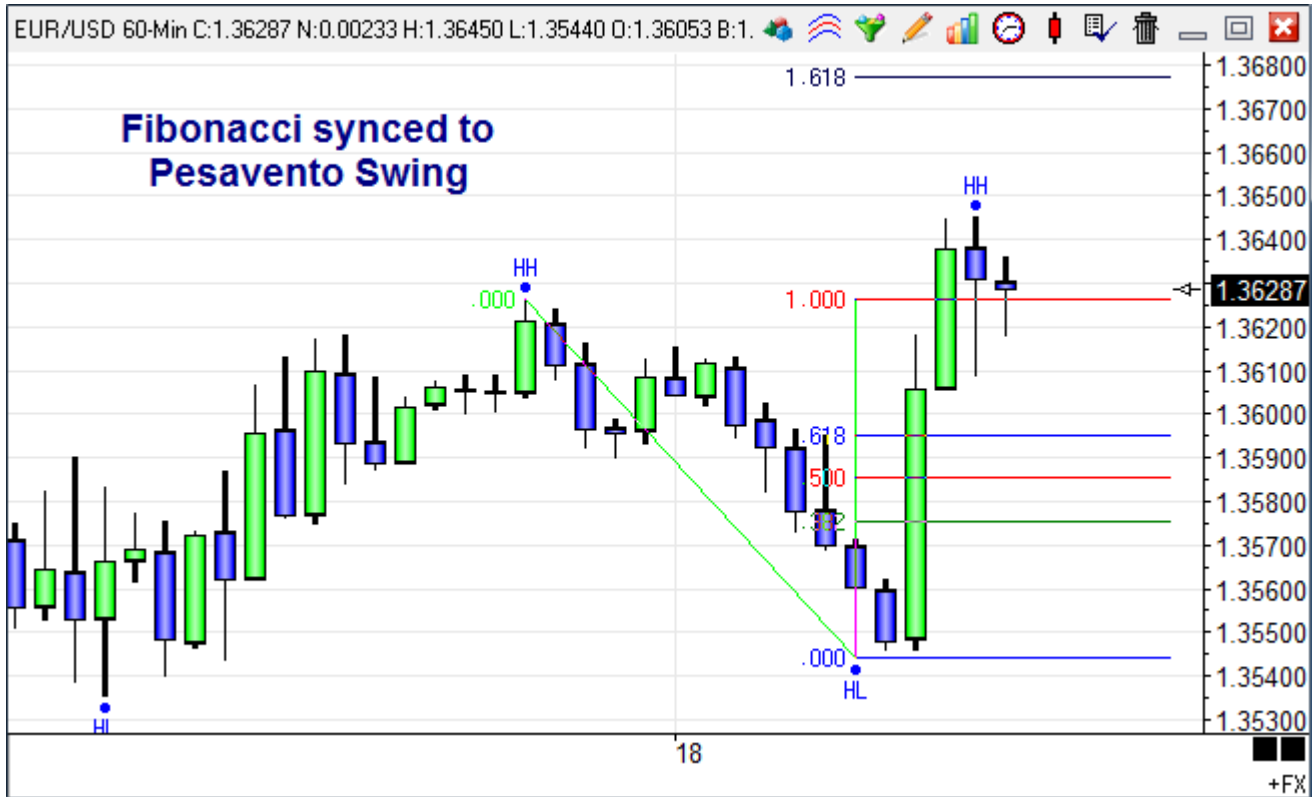
Set Param 1, 2, 3, U/D, and L/R have one parameter and that is the value to assign. Parameter 1 for a Moving Average study would be the 1st average parameter on the study property form. U/D is the up / down price adjustment. L/R is the horizontal offset adjustment.

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

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 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.

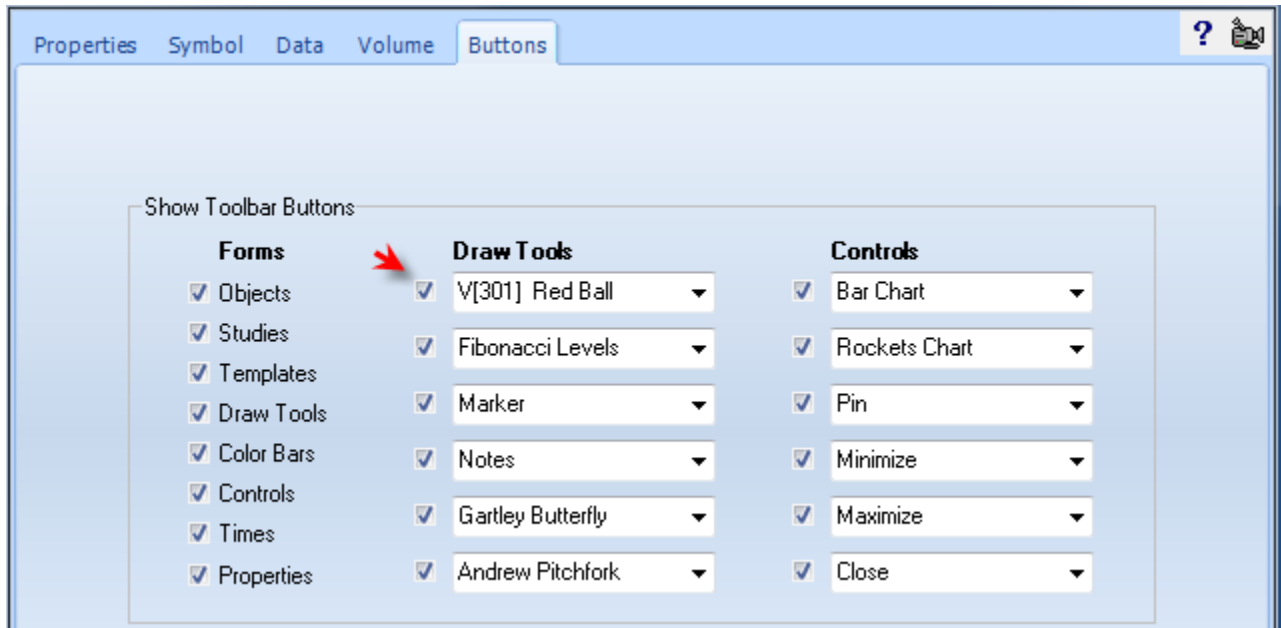


Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4
Study	0	= Pesavento	g	Swing bar
<div style="display: flex; align-items: center; gap: 10px;"> ↩ → ↻ </div>				
A	Study	[0] := Pesavento.Swing bar		<input type="checkbox"/>
B	Branching	if Not [A] then abort execution		<input type="checkbox"/>
C	Study	[PointAprice] := Pesavento.Swing 4 value		<input type="checkbox"/>
D	Study	[PointBprice] := Pesavento.Swing 3 value		<input type="checkbox"/>
E	Study	[PointCprice] := Pesavento.Swing 2 value		<input type="checkbox"/>
F	Study	[PointAindex] := Pesavento.Swing 4 index		<input type="checkbox"/>
G	Study	[PointBindex] := Pesavento.Swing 3 index		<input type="checkbox"/>
H	Study	[PointCindex] := Pesavento.Swing 2 index		<input type="checkbox"/>
I	Study	[Handle] := Andrews.Handle		<input type="checkbox"/>
J	Action	object([Handle]).Set Point A([PointAindex], [PointAprice])		<input type="checkbox"/>
K	Action	object([Handle]).Set Point B([PointBindex], [PointBprice])		<input type="checkbox"/>
L	Action	object([Handle]).Set Point C([PointCindex], [PointCprice])		<input type="checkbox"/>

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

Refresh Button

Chart toolbars have 6 buttons that can be configured on the chart's property form. Normally these buttons are customized to activate placement of Draw Tools. However, there are selections for assigning a True value to variables [301] through [305]. This example will configure one of the buttons to refresh the chart for today.



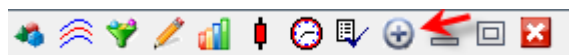
On the Buttons tab of the chart properties, change one of the Draw Tools buttons to the V[301] selection. This button will then change variable [301] to True when clicked.

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Action	0	= if #2 then Button[#].Image :=	1	[301]	0
		[182		183	0
A	Action	if [301] then Button1.Image := 182 else 183			
B	Action	if ([301]) then Action 50			
C	Expression	[301] := False			

A – Change the image of the 1st Draw Tool button to image 182 when [301] is True, and show image 183 when [301] is false. See the [Image List](#) in the Appendix for the image indexes.

B – When [301] is true the Action 50 is performed, which is the Refresh 1st Quantity. See the documentation for [if ## then Action #2](#).

C – Variable [301] is set true by clicking the toolbar button. Therefore, reset it back to False.



After clicking the circled plus button, the next tick will execute the DYO and refresh the chart. The button will change to the circled minus image until the next tick update.

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.

Spreadsheet Properties

Regular Cells		Fixed Cells	
<input type="checkbox"/> Font Color	Decimal Format	<input checked="" type="checkbox"/> Font Color	
<input checked="" type="checkbox"/> Grid Color	2	<input type="checkbox"/> Grid Color	
<input type="checkbox"/> Cell Color		<input type="checkbox"/> Cell Color	

Bands

<input checked="" type="checkbox"/> First Band Color	1	Rows	<input checked="" type="checkbox"/> Enable Bands
<input checked="" type="checkbox"/> Second Band Color	5	Rows	

Color Theme

Save Font

Calculate

Mode <input type="radio"/> Manual <input type="radio"/> On Cell Change <input checked="" type="radio"/> Timer	10 Number of Columns	6 Number of Rows	Style <input checked="" type="radio"/> A1 <input type="radio"/> RxCy
	3	Timer Frequency 1/3rd second	

The next image shows the cell formula used.

	ES #F	B	C	D	E	F	G	H	I	J
1	=Last("ES #F","ES")	High	Low	prior High	prior Low	STO %K	%K/%D	RSI	RSI/AVE	TREND
2	5 min bar	=Get(100)	=Get(101)	=Get(102)	=Get(103)	=Get(104)	=IF(Get(105),"Above","Below")	=Get(106)	=IF(Get(107),"Above","Below")	↑
3	60 min bar	=Get(110)	=Get(111)	=Get(112)	=Get(113)	=Get(114)	=IF(Get(115),"Above","Below")	=Get(116)	=IF(Get(117),"Above","Below")	↓
4	Daily	=Get(120)	=Get(121)	=Get(122)	=Get(123)	=Get(124)	=IF(Get(125),"Above","Below")	=Get(126)	=IF(Get(127),"Above","Below")	↑
5	Weekly	=Get(130)	=Get(131)	=Get(132)	=Get(133)	=Get(134)	=IF(Get(135),"Above","Below")	=Get(136)	=IF(Get(137),"Above","Below")	↑
6	Monthly	=Get(140)	=Get(141)	=Get(142)	=Get(143)	=Get(144)	=IF(Get(145),"Above","Below")	=Get(146)	=IF(Get(147),"Above","Below")	↑

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	I
%K/%D	RSI/AVE
=IF(Get(105),"Above","Below")	=IF(Get(107),"Above","Below")
=IF(Get(115),"Above","Below")	=IF(Get(117),"Above","Below")
=IF(Get(125),"Above","Below")	=IF(Get(127),"Above","Below")
=IF(Get(135),"Above","Below")	=IF(Get(137),"Above","Below")
=IF(Get(145),"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	<input type="checkbox"/>		
B	Expression	[101] := Low	<input type="checkbox"/>		
C	Expression	[102] := Prior High	<input type="checkbox"/>		
D	Expression	[103] := Prior Low	<input type="checkbox"/>		
E	Study	[104] := STO.%K value	<input type="checkbox"/>		
F	Study	[105] := STO.%K >= %D	<input type="checkbox"/>		
G	Study	[106] := RSI.RSI value	<input type="checkbox"/>		
H	Study	[107] := RSI.RSI >= Average	<input type="checkbox"/>		
I	Study	[108] := Trend.Swing direction	<input type="checkbox"/>		
J	Action	if [108] then Marker(10, 2) := Marker	<input type="checkbox"/>	↑	
K	Action	if [108] = 0 then Marker(10, 2) := Marker	<input type="checkbox"/>	↓	

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.

[#]	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

```

--- Spreadsheet ---
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 #2 oper 0 then ColWidth( #3 ) := #4
if #2 oper 0 then RowHeight( #3 ) := #4
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)
    
```

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	<input type="checkbox"/>	
		{ 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.

Trading System

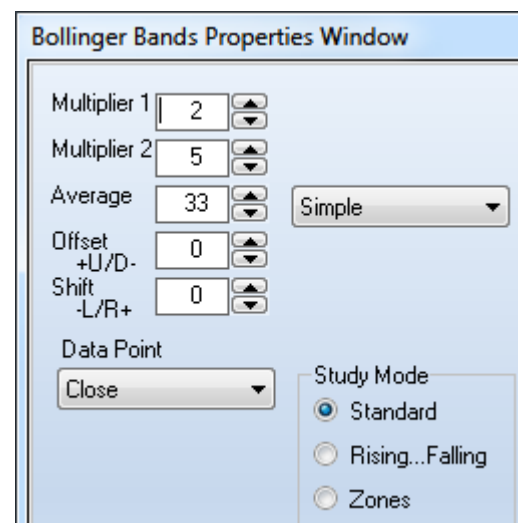
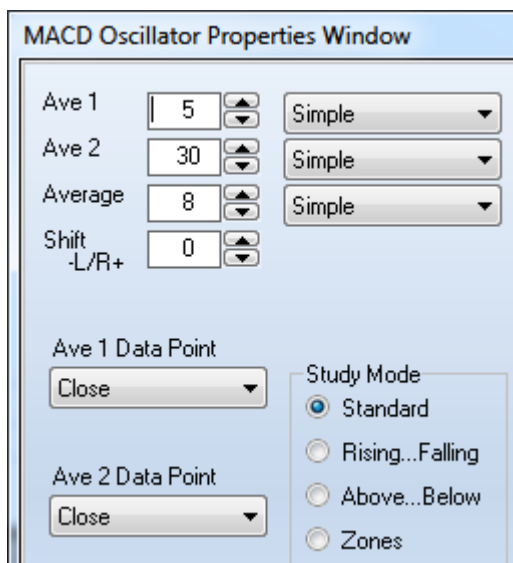
The following example introduces statements for building a trading system and back testing the results. No claim is being made that this example will be profitable in the future.

The example is simple for the sake of being a teaching tool, yet complex enough to illustrate a variety of trading statements. The example uses the following:

- A MACD study will be used to signal when to place a Buy Stop.
- A Bollinger band study will be used for entry and exit stops.
- A DYO study implements the trading logic and displays the stop placements.
- The DYO sets a Buy Stop, Sell Limit for a scalp objective, and a trailing Sell Stop.

The MACD study parameters are 5, 35, and 8. The lines are not plotted.

The Bollinger band parameters are 2, 5, 33. The lines are not plotted.



Rules:

Place a Buy Stop at the upper Bollinger band when the MACD is above zero and above its average.

At the same time, place a Sell Limit order at a scalp objective above the upper Bollinger band.

When a long position is taken, place a trailing Sell Stop at the lower Bollinger band and move this stop as the lower band changes.

For the sake of simplicity, this example only shows the rules for a long position. The inverse logic would be applied for entering and managing short positions.

System Implementation:

The system rules are implemented by the following Design Your Own (DYO) study.

Study Name MACD Bollinger	Study Location Chart	Marker Location Study Value	Label Location None	Grid Tab Default	<input type="checkbox"/> Use as Default			
Study Scale Chart Scale	Variables File Ensign	Message Location None	<input type="checkbox"/> Draw Behind Bars					
Message Text	Font	True	False	Panel	<input type="checkbox"/> Privatize			
					<input type="checkbox"/> Close Only			
Sound	Trading System				<input type="checkbox"/> Show Values			
<input checked="" type="radio"/> Silent	<input type="radio"/> Beep	<input type="radio"/> Voice	<input type="radio"/> WAV	<input type="checkbox"/> Sound once per bar	<input type="checkbox"/> Email			
WAV file	Price Last	Quantity 1	Commission 0	<input type="checkbox"/> Auto Remove				
				<input checked="" type="checkbox"/> Auto Completion				
Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
Study	26 MACD	= MACD 5,30 HH		MACD value	0	<input type="checkbox"/>		
					0	<input type="checkbox"/>		
A	Study	[MACD] := MACD.MACD value				<input type="checkbox"/>		
B	Study	[MACDave] := MACD.Average value				<input type="checkbox"/>		
C	Study	[Bollinger Upper] := BOL.1st Upper band				<input type="checkbox"/>		
D	Study	[Bollinger Lower] := BOL.1st Lower band				<input type="checkbox"/>		
E	Expression	[28] := ([MACD] > 0) AND ([MACD] > [MACDave])		ZON		<input type="checkbox"/>		
F	Flag	[Rule1b] := [E] goes True		ZON		<input type="checkbox"/>		
G	Action	if ([Rule1b] AND Position is Flat) then Buy Stop Price := [Bollinger Upper]				<input type="checkbox"/>		
H	Expression	[Long Scalp] := [Bollinger Upper] + 0.003				<input type="checkbox"/>		
I	Action	if ([Rule1b] AND Position is Flat) then Sell Limit Price := [Long Scalp]				<input type="checkbox"/>		
J	Action	if (Position is Long) then Sell Stop Price := [Bollinger Lower]	<input checked="" type="checkbox"/>		[D] Value	<input type="checkbox"/>		
K	Expression	[0] := Sell Limit Price	<input checked="" type="checkbox"/>		Study Value	<input type="checkbox"/>		
L	Expression	[0] := Buy Stop Price	<input checked="" type="checkbox"/>		Study Value	<input type="checkbox"/>		

Row A and B read MACD study values into variables for use on Row E.

Row C and D read Bollinger study values into variables for use on Rows G, H and J.

Row E tests for the MACD value being above zero and being above its average.

Row F is the signal. When Row E changes from False to True, place a Buy Stop which is done by Row G. The location of the buy stop is the upper Bollinger band price.

Row H calculates the scalp objective by adding 30 pips to the Buy Stop price.

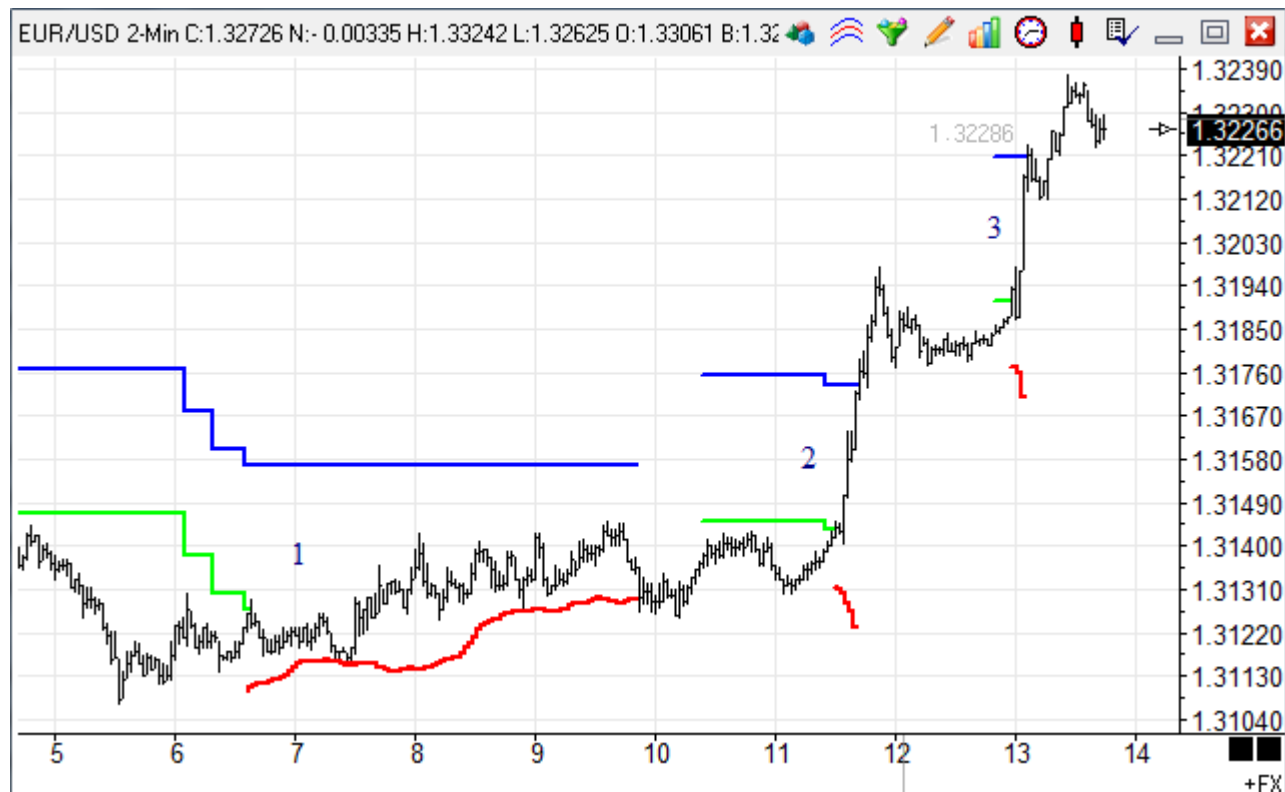
Row I uses the same Row F signal and sets the Sell Limit at the scalp objective.

Row J will place and move a Sell Stop when the trade position is Long. This stop will be the lower Bollinger band value. Row J also plots a red line at this stop value from Row D.

Row K plots a blue line at the Sell Limit which is the scalp objective.

Row L plots a green line at the Buy Stop, and a price penetration of this stop initiates a long position.

Chart Example:



This section of the chart shows 3 trades, the 1st of which was stopped out at a small gain. Trades #2 and #3 are the break out trades the system seeks, which achieved the scalp objective of 30 pips.

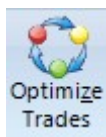
Trade Results:

Click the Trade Report drop down menu on the Account button on the Main ribbon to see a ledger of trade results, such as this example for the EUR/USD.2 minute chart.

Symbol	Pos	Quantity	EntryDate	EntryPrice	ExitPrice	Profit/Loss	Balance	Notes
EUR/USD	L	1	12-28-10	1.31125	1.31007	\$ -118.00	\$ 717.00	18:22 - 18:40
EUR/USD	L	1	12-28-10	1.31121	1.31246	\$ 125.00	\$ 842.00	20:44 - 00:02
EUR/USD	L	1	12-29-10	1.31271	1.31347	\$ 76.00	\$ 918.00	01:00 - 03:24
EUR/USD	L	1	12-29-10	1.31269	1.31291	\$ 22.00	\$ 940.00	06:38 - 09:52
EUR/USD	L	1	12-29-10	1.31434	1.31734	\$ 300.00	\$ 1,240.00	11:30 - 11:42
EUR/USD	L	1	12-29-10	1.31907	1.32207	\$ 300.00	\$ 1,540.00	12:58 - 13:06
EUR/USD	L	1	12-29-10	1.32254	1.32209	\$ -45.00	\$ 1,495.00	15:10 - 17:36
EUR/USD	L	1	12-29-10	1.32240	1.32540	\$ 300.00	\$ 1,795.00	19:12 - 20:06
EUR/USD	L	1	12-29-10	1.32464	1.32433	\$ -31.00	\$ 1,764.00	20:08 - 21:12

The 3 trades shown on the chart are rows 18, 19, and 20 in the Trade Report.

Optimize Trades



Documentation of the Optimize Trades feature will continue with the previous example which initiates a Long position with a Buy Stop at an upper Bollinger band, and seeks to exit on a profit objective. The trade is protected by a trailing stop at the lower Bollinger band level.

Though the example looks promising with favorable trade results, one does not know if the parameters chosen are optimal. For example, would more profit be achieved by trying to scalp a larger move? Or, should the Bollinger bands used for entry and exit be closer to the price action, or farther away?

Ensign has a powerful tool that can be used to research these questions. Click the Optimize Trades button on the Setup ribbon to show the following form.

The screenshot shows the 'Optimizer' window with the 'Setup' tab selected. The interface includes a dropdown menu for the 'Optimizer control file' set to 'MACD Bollinger', and buttons for 'Save', 'Run Optimizer', and 'Clear Form'. Below this is a 'List of Layouts or Charts' area. To the right, there are checkboxes for 'Evaluate each layout or chart on the list' and 'Results print method'. A table lists optimization loops with columns for 'Title', 'From', 'To', 'Step', and 'Assign Variable'. The 'Optimize' section has radio buttons for 'Trade Profits' (selected), 'Minimize Variable', and 'Maximize Variable', along with a '0' value in a dropdown. A 'Load Template' dropdown is set to 'JaysonOptimize', and 'Execute Script' is checked. The bottom right corner displays '3 x 3 x 8 = 72 iterations'.

	Title	From	To	Step	Assign Variable
<input checked="" type="checkbox"/> Enable LoopA	MACD	25	35	5	
<input checked="" type="checkbox"/> Enable LoopB	BOL	1.8	2.2	0.2	
<input checked="" type="checkbox"/> Enable LoopC	Scalp	0.0020	0.0034	0.0002	50

Iterations

The Setup tab is used to define the iterations that are to be tested. This example will vary the parameter for the 2nd MACD average from 25 to 35 in increments of 5. The width multiplier parameter for the Bollinger bands will be tested in the range of 1.8 to 2.2 in increments of 0.2. And the scalp objective will be varied from 20 pips to 34 pips in 2 pip increments. These 3 parameters are nested loops, where LoopC is the inner most loop.

The MACD loop tests 3 values. The BOL loop tests 3 values. And the Scalp loop tests 8 variations. The total number of combinations examined will be $3 \times 3 \times 8 = 72$.

Results

	Chart	MACD	BOL	Scalp	Win #	Loss #	Total #	% Win	\$ Win	\$ Loss	Profit	Ratio	Average
35	EUR_USD.2	30.00	1.8000	0.0032	32	24	56	57	5812	-2033	3780	2.8596	67.50
36	EUR_USD.2	30.00	1.8000	0.0034	31	25	56	55	4877	-2162	2715	2.2559	48.48
38	EUR_USD.2	30.00	2.0000	0.0020	32	27	59	54	5298	-2396	2903	2.2115	49.20
39	EUR_USD.2	30.00	2.0000	0.0022	31	27	58	53	5579	-2474	3105	2.2553	53.53
40	EUR_USD.2	30.00	2.0000	0.0024	30	26	56	54	5634	-2462	3172	2.2887	56.65
41	EUR_USD.2	30.00	2.0000	0.0026	28	26	54	52	5554	-2380	3174	2.3337	58.78
42	EUR_USD.2	30.00	2.0000	0.0028	28	25	53	53	5750	-2154	3595	2.6689	67.84
43	EUR_USD.2	30.00	2.0000	0.0030	28	24	52	54	5886	-1973	3913	2.9831	75.24
44	EUR_USD.2	30.00	2.0000	0.0032	27	24	51	53	5048	-2104	2944	2.3994	57.73
45	EUR_USD.2	30.00	2.0000	0.0034	27	24	51	53	5308	-2104	3204	2.5229	62.83
47	EUR_USD.2	30.00	2.2000	0.0020	30	24	54	56	5342	-2570	2771	2.0782	51.32

The table contains the trade summaries for the 72 tests made. The view shown above has been positioned to show the settings that generated the highest Profit, which is Row 43. For the test sample which spanned a 7 day period at the end of 2010, the optimal parameters would have been to use a 30 period average for the 2nd MACD average, a 2.0 multiplier for the Bollinger bands, and a scalp objective of 30 pips.

Another use for the table of results is to evaluate how sensitive a system's design might be the parameters that are used. For example, compare the results of rows 38 through 41. The scalp parameter is varied from 20 pips to 26 pips, but the Profit remained quite steady through that range of scalp values.

File of Optimal Parameters

The optimal parameters found for Loop A, Loop B, and Loop C are written to an ASCII file in the C:\Ensign10\Optimizer\ folder. The file name used is the chart's filename with a suffix of '.txt' appended. The file will contain the values for Loops A, B and C separated by commas.

Example filename: C:\Ensign10\Optimizer\EUR_USD.2.txt

Example content: 30.0000, 2.0000, 0.0030

A DYO can read these values with a function. Function | Optimizer File Field(#2)

The #2 value should be 1, 2 or 3 to select the 1st, 2nd, or 3rd field to read from this file.

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Function	5 Scalp	= Optimizer File Field(#2)		2	0
		{			0
A	Function	[Scalp] := Optimizer File Field(2)			

Optimizer Running

An improvement to the design would be to make the DYO automatically adapt itself for when it is run normally, and for when it is controlled by the Optimizer. A row could have been added that conditionally sets [50] to 0.003 when the Optimizer is not running. The statement would be:

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4
Action	50	= if ## then V := #2		0.003
		Not	[Optimizer Running	
A Action if (Not Optimizer Running) then [50] := 0.003				

Such a statement would permit the the DYO to be run normally or by the Optimizer. The Optimizer Running flag will be True when the Optimizer is running.

The next example shows how normal assignments for several Variables can be skipped when the Optimizer is running. Add an initialization DYO as the first DYO in a design, like this:

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4
Branching	0	= if # then abort execution		Optimizer Running
A Branching if Optimizer Running then abort execution				
B	Expression	[50] := 0.003		
C	Expression	[51] := 5		
D	Expression	[67] := 27		

The Branching statement aborts this DYO's execution when the Optimizer is running. When the Optimizer is running, it is the Optimizer that should assign its loop values to the variables. The variables would be assigned by the DYO when the Optimizer is not running.

Summary

Designing a profitable trading system can be rewarding and yet challenging. Many advanced users find the chase to be a lot of fun. Ensign has the tools which can be used to help you succeed in your research. As demonstrated in this example, DYOs can be used to implement the rules of a trading system. The program will accomplish the trades after stop and limit prices are set. And the DYO can execute Buy and Sell commands, and exit a position.

The trades that are made by the system post to a Trade Report which shows the details of the entry and exit prices, dates, times, and profits. The Optimizer is an additional power tool that can be used to research which parameters should be used in the system's design.

The Ensign team hopes you enjoy the power and flexibility available to you. Be creative and experience the exhilaration of researching and designing your own trading systems.

Trend Runner System

The quest to find a trading system that will help put money in the bank can be difficult and frustrating. Often what has worked in the past might not work now. Though a system might be sound, its parameters may just need to be adjusted. The more complex a trading system's design, the more variations in settings where some might yield a higher win percentage while others have a higher per trade average.

Designing a trading system takes a lot of work and testing, testing, testing. The Optimizer in Ensign 10 shortens the research time by rapidly testing a large variety of parameter settings. The Optimizer can perform hundreds of iterations and evaluate settings that otherwise might not have been considered. Lets see how the Optimizer can improve a trading system.

Of course it needs something to optimize. The following Trend Runner system is designed around a trailing stop. No implication is made that this system makes money, or that it works in all markets or with all time frames. It is just an example to illustrate designing a system, using the Optimizer, dressing the chart with information, and emailing notifications of system events.



The system uses a trailing stop whose value is the highest high or lowest low of the last N number of bars. And there is a bias for being long or short based on a moving average. Thus there are two study parameters that are worthy of testing different variations. One is the number of bars in the look back period for the trailing stop, and the other is the parameter for the moving average that determines when the trades are entered.

The Trend Runner system requires several DYO's to implement. Some of the DYO's implement the trade rules, some create the 'eye candy' for showing trade results, and some send email notifications. Though the system looks complex, it is worthy of being studied to understand how a trading system can be designed and optimized. Lets begin.

- Study Settings
- Trades
- Profit Curve
- Grid Lines for Window 5
- At Risk
- Calculate Stats
- Show Stats
- Fill Markers
- Trade Markers
- Email Move SELL Stop
- Email Move BUY Stop
- Email Profit Target
- Email BUY/SELL Entry

Study Settings

This DYO initializes variables and calculates the exit stops and the moving average.

Study Name: Study Location: Marker Location: Label Location: Grid Tab:

Study Scale: Variables File: Message Location:

Message Text: Font: True False Panel

Sound: Silent Beep Voice WAV Sound once per bar WAV file:

Trading System: Price: Quantity: Commission:

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
Branching	0	= if # then go ahead [#] rows	4	Optimizer Running	0	<input type="checkbox"/>		
<div style="border: 1px solid #ccc; padding: 2px;"> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> </div>								
A	Branching	if Optimizer Running then go ahead 4 rows				<input type="checkbox"/>		
B	Expression	[Stop Period] := 75				<input type="checkbox"/>		
C	Expression	[Ave Period] := 16				<input type="checkbox"/>		
D	Expression	[Send Email] := False				<input type="checkbox"/>		
E	Expression	[Profit Target] := 100				<input type="checkbox"/>		
F	Expression	[Leverage] := 100000				<input type="checkbox"/>		
G	Function	[Upper Stop] := Highest(High, [Stop Period])				<input type="checkbox"/>		
H	Function	[Lower Stop] := Lowest(Low, [Stop Period])				<input type="checkbox"/>		
I	Function	[Average] := Simple Average(Close, [Ave Period])				<input checked="" type="checkbox"/>		 Study Value
J						<input type="checkbox"/>		
K	Expression	[UpperMoved] := [G] < [G][-1]				<input type="checkbox"/>	ZON	
L	Expression	[LowerMoved] := [H] > [H][-1]				<input type="checkbox"/>	ZON	

- Use as Default
- Plot Behind Bars
- Privatize
- Close Only
- Show Values
- Email
- Auto Remove
- Auto Completion

A – During normal operation, the Optimizer is not running, and therefore, execution will flow to row B. When the Optimizer is running, row B and C need to be skipped because the Optimizer will set the Stop Period value and the Ave Period value.

69

- D – This variable is a flag for enabling or disabling the sending of email notifications.
- E – Set a Profit Target for the purpose of sending an email when this profit has been reached.
- F – The leverage is used to convert PIPS to Dollars.
- G & H - Find the highest high or lowest low of the last N bars. N has been stored in the Stop Period variable so that the Optimizer can change it. In order for the operator field to read a variable the index needs to be in the range of 200 through 230. The index for the Stop Period variable is 202.
- I – Calculate a simple average, where the period parameter is stored in a variable so it can be changed by the Optimizer. The Ave Period parameter is stored in variable 203. The average determines when a trade is entered depending on its relationship to the high or low of the bar.
- K – This Boolean flag will be True when the row G Upper Stop is lowered. This flag controls sending an email notification that the Buy Stop has been moved.
- L – This Boolean flag will be True when the row H Lower Stop is raised. This flag controls sending an email notification that the Sell Stop has been moved.

Trades

This DYO implements the rules for buying, selling, and exiting trades.

A	Action	if Position is Flat AND (Low >= [Average]) then Buy Market	<input type="checkbox"/>				
B	Action	if Position is Flat AND (High <= [Average]) then Sell Market	<input type="checkbox"/>				
C			<input type="checkbox"/>				
D	Action	if (Low <= [Lower Stop]) then Exit Long Position	<input type="checkbox"/>				
E	Action	if (High >= [Upper Stop]) then Exit Short Position	<input type="checkbox"/>				
F			<input type="checkbox"/>				
G	Action	[0] := [Lower Stop]; if (Position is Long) then plot V	<input checked="" type="checkbox"/>		Study Value	[Lower Stop]	
H	Action	[0] := [Upper Stop]; if (Position is Short) then plot V	<input checked="" type="checkbox"/>		Study Value	[Upper Stop]	

- A – Buy at the Market when the Position is Flat and the bar's Low is above the Average.
- B – Sell at the Market when the Position is Flat and the bar's High is below the Average.
- D – Exit a Long Position when the bar's Low touches the trailing Lower Stop.
- E – Exit a Short Position when the bar's High touches the trailing Upper Stop.
- G – Plot the Lower Stop and show its current value when the Position is Long.
- H – Plot the Upper Stop and show its current value when the Position is Short.

The Label Location for the stop values shown by row G and H is the Margin 2 selection.

This DYO has its Close Only box checked so the evaluation is done at the end of the bar, and the bar's Close price is used as the trade's execution price. The trade quantity is 1.

Profit Curve

This DYO shows the Profit Curve in the sub-window and labels the current total. This DYO is 'eye candy' that shows the ebb and flow of the profit. This DYO could be added to other trade systems that executed their trades using DYO statements to buy, sell and exit positions.

Study Name Profit Curve	Study Location Window 5	Marker Location Study Value	Label Location Margin 2	Grid Tab Default	<input type="checkbox"/> Use as Default			
Study Scale Data Set +5%		Variables File -TrendRunner	Message Location None		<input type="checkbox"/> Plot Behind Bars			
Message Text	Font	True	False	Panel	<input type="checkbox"/> Privatize			
					<input checked="" type="checkbox"/> Close Only			
Sound <input checked="" type="radio"/> Silent <input type="radio"/> Beep <input type="radio"/> Voice <input type="radio"/> WAV <input type="checkbox"/> Sound once per bar	Trading System Price Close				<input checked="" type="checkbox"/> Show Values			
WAV file Browse	Quantity 1	Commission 0			<input type="checkbox"/> Email			
Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
Expression	19 Profit	= Total Dollars	+	Profit Dollars	0	<input checked="" type="checkbox"/>		
		(0			
A	Expression	[Profit] := Total Dollars + Profit Dollars		<input checked="" type="checkbox"/>			Paint Color	
B	Expression	[0] := [Profit] < 0		<input checked="" type="checkbox"/>			Paint Color	
C	Expression	[0] := [Profit]		<input checked="" type="checkbox"/>			Study Value	
D	Action	[0] := 0; if (True) then plot V		<input checked="" type="checkbox"/>			Study Value	
E	Expression	[0] := [Profit]		<input checked="" type="checkbox"/>			Study Value	\$([Profit.])

A – The total profit is the sum of the Total Dollars in the closed positions plus the Profit Dollars in the current open trade (if any). This line also initializes the paint can color to green.

B – Change the paint can color to red if the Profit is less than zero.

C – Plot the Profit using the Fill marker with Black as the color selected. Black is the special case color that will use the color in the paint can set by row A or row B. Thus the fill color is either green or red. The fill is between the Profit value on row C and the value on row D.

D – This statement's value is a zero so the row C fill will color both positive and negative fills to zero on the sub-window scale.

E – Plot the Profit curve with a thick blue line, and post its value in the Margin 2 position.

The Grid Lines object for sub-window 5 can be configured to show more or fewer grid lines, and control the grid line style, colors, and labels.

At Risk

This DYO calculates and shows the information in the first 2 sections below the chart.

A	Expression	[At Risk] := 0	<input type="checkbox"/>				
B	Branching	if Position is Long then do next row	<input type="checkbox"/>				
C	Expression	[At Risk] := (Long Entry Price - [Lower Stop]) * [Leverage]	<input type="checkbox"/>				
D	Branching	if Position is Short then do next row	<input type="checkbox"/>				
E	Expression	[At Risk] := ([Upper Stop] - Short Entry Price) * [Leverage]	<input type="checkbox"/>				
F	Action	if ([At Risk] < 0) then [At Risk] := 0	<input type="checkbox"/>				
G	Expression	[0] := 0	<input checked="" type="checkbox"/>		Study Value	1 At Risk,\$[At	
H	Expression	[ProfitDollars] := Profit Dollars	<input checked="" type="checkbox"/>		Study Value	2 Current	
I	Action	if (Position goes Long OR Position goes Short) then [Index] := Bar Index	<input type="checkbox"/>				
J	Expression	[Profit Period] := Bar Count - [Index]	<input type="checkbox"/>				
K	Function	[HighestHigh] := Highest(High, [Profit Period])	<input type="checkbox"/>				
L	Function	[LowestLow] := Lowest(Low, [Profit Period])	<input type="checkbox"/>				

A – A variable named At Risk is initialized to zero.

B & C – If the Position is Long, At Risk is calculated using the Long Entry Price and the Lower Stop.

D & C – If the Position is Short, At Risk is calculated using the Short Entry Price and the Upper Stop.

F – If At Risk is a negative value, it means the Stops have moved beyond the entry prices and locked in a profit. The At Risk is therefore set to zero.

G – This row posts the text and the At Risk value in section 1 on a red background.

H – This row posts the text and the Current Trade value in section 2 on a blue background.

Rows I, J, K and L are making calculations for finding two values for the current trade: its Maximum Draw Down and its Maximum Profit.

I – Initialize a variable with the bar index when a new trade position is taken.

J – Calculate the number of bars since a position was taken. This is the look back period.

K – Find the HighestHigh in the look back period since the trade was taken.

L – Find the LowestLow in the look back period since the trade was taken.

Values from rows K and L will be used in the next DYO to Calculate Stats.

Calculate Stats

This DYO will calculate the Draw Down, the Max Profit, and the Locked Profit for the current trade.

A	Branching	if Not Position is Long then go ahead 5 rows	<input type="checkbox"/>
B	Expression	[Draw Down] := (Long Entry Price - [LowestLow]) * [Leverage]	<input type="checkbox"/>
C	Expression	[Max Profit] := ([HighestHigh] - Long Entry Price) * [Leverage]	<input type="checkbox"/>
D	Expression	[Locked Profit] := ([Lower Stop] - Long Entry Price) * [Leverage]	<input type="checkbox"/>
E			<input type="checkbox"/>
F	Branching	if Not Position is Short then go ahead 5 rows	<input type="checkbox"/>
G	Expression	[Draw Down] := ([HighestHigh] - Short Entry Price) * [Leverage]	<input type="checkbox"/>
H	Expression	[Max Profit] := (Short Entry Price - [LowestLow]) * [Leverage]	<input type="checkbox"/>
I	Expression	[Locked Profit] := (Short Entry Price - [Upper Stop]) * [Leverage]	<input type="checkbox"/>
J			<input type="checkbox"/>
K	Action	if ([Locked Profit] < 0) then [Locked Profit] := 0	<input type="checkbox"/>

A – Skip down to row F unless the Position is Long. Row B, C and D are calculations for a Long Position.

B – The maximum Draw Down is the spread from the Entry Price to the LowestLow.

C – The maximum Profit is the spread from the HighestHigh to the Entry Price.

D – Locked Profit is the spread between the Lower Stop and the Entry Price.

F – Skip down to row K unless the Position is Short. Row G, H and I are calculations for a Short Position.

G – The maximum Draw Down is the spread from the HighestHigh and the Entry Price.


H – The maximum Profit is the spread from the Entry Price and the LowestLow.

I – Locked Profit is the spread between the Entry Price and the Upper Stop.

K – If the Locked Profit value is less than zero, reset it to zero. Only positive values represent a profit that has been locked in by the exit stop being beyond the entry price.

Show Stats

This DYO shows titles in sections 3, 4 and 5, and the values calculated by the prior DYO.

A	Expression	[0] := 0	<input checked="" type="checkbox"/>		Study Value	3 Max Draw
B	Expression	[0] := 0	<input checked="" type="checkbox"/>		Study Value	4 Trade
C	Expression	[0] := 0	<input checked="" type="checkbox"/>		Study Value	5 Locked

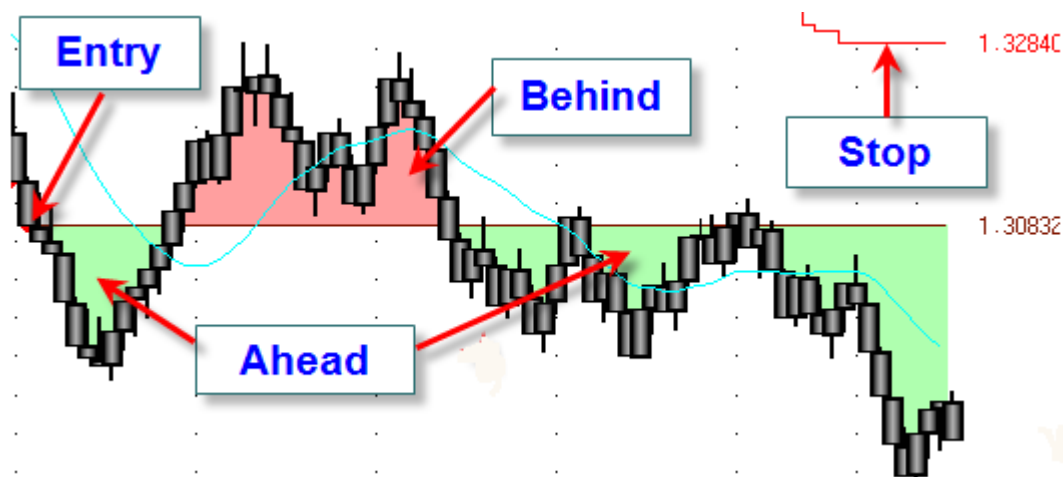
A – The row label posts the title and the value. Its full entry looks like this:

|3|Max Draw Down,\$[13.2]

|3| is section 3. 'Max Draw Down' is the title. Comma starts a 2nd line. \$ is text. [13.2] is variable [13] which is the Draw Down variable, and .2 formats the value to show 2 decimals.

Fill Markers

The next DYO is more 'eye candy' which colors the zone between the bars and the trade entry price. Profitable positions are colored green, and losing positions are colored red.



This example shows when the short position is profitable (green fill) and behind (red fill). The exit stop is trailing behind and recently lowered to 1.32840. The short entry price is 1.30832.

A	Branching	if Position is Flat then abort execution	<input type="checkbox"/>		
B	Action	if Position is Long then [TradePrice] := Long Entry Price else [TradePrice]	<input type="checkbox"/>		
C	Expression	[0] := Profit Points >= 0	<input checked="" type="checkbox"/>		Paint Color
D	Expression	[0] := Profit Points < 0	<input checked="" type="checkbox"/>		Paint Color
E	Expression	[0] := [TradePrice]	<input checked="" type="checkbox"/>		Study Value
F	Expression	[0] := Close	<input type="checkbox"/>		

A – Exit this DYO when the Position is Flat. There is nothing to do when flat.

B – If the Position is Long then the TradePrice is the Long Entry Price. If the Position is Short the TradePrice is the Short Entry Price.

C – Initialize the paint can color to green when the trade is profitable.

D – Initialize the paint can color to red when the trade is losing money.

E – Fill the zone between 2 prices using the paint can color. The 2 prices for the fill are the TradePrice on row E and the bar's Close price on the next row which is row F.

This DYO has the Plot Behind Bars box checked so the bars show on top of the fill.

Trade Markers

This DYO adds more 'eye candy' to show the entry price, entry and exit points, and a color bar on the bottom of the screen representing the direction of the trade. The bar is green for a Long position and red for a Short position.

A	Expression	[0] := Long Entry Price	<input checked="" type="checkbox"/>	—		Study Value	[A]
B	Expression	[0] := Short Entry Price	<input checked="" type="checkbox"/>	—		Study Value	[B]
C			<input type="checkbox"/>				
D	Expression	[0] := Position goes Flat	<input checked="" type="checkbox"/>	×		Last	
E	Expression	[0] := Position goes Long	<input checked="" type="checkbox"/>	●		Last	
F	Expression	[0] := Position goes Short	<input checked="" type="checkbox"/>	●		Last	
G	Expression	[0] := Position is Long	<input checked="" type="checkbox"/>			Bottom Row	
H	Expression	[0] := Position is Short	<input checked="" type="checkbox"/>			Bottom Row	
I			<input type="checkbox"/>				
J	Expression	[0] := Position is Long	<input checked="" type="checkbox"/>			Study Value	
K	Expression	[0] := Position is Short	<input checked="" type="checkbox"/>			Study Value	

A – Plot a line at the Long Entry Price in dark green and show its value in the margin.

B – Plot a line at the Short Entry Price in dark red and show its value in the margin.

D – Place a X marker on the bar's Last price when a trade is exited.

E – Place a green dot marker on the bar's Last price when a Long position is taken.

F – Place a red dot marker on the bar's Last price when a Short position is taken.

G – Plot a fat green bar on the bottom of the chart while the Position is Long.

H – Plot a fat red bar on the bottom of the chart while the Position is Short.

J & I – Color Bar the candlestick body green or red depending on the position being Long or Short. The yellow highlight on the DYO form indicates the lines are currently commented out and not executed. But they could be activated by selecting Expression in the Category drop down box. They are included in the example to show how the candlesticks could be colored based on the position. The Color Bars study selection must be on the Normal selection so that Color Bars does not compete with the DYO in coloring the candles.

Email Notifications

The remaining DYO's send email when alert conditions are triggered. Email can be enabled or disabled with the Send Email variable in the Study Settings DYO. All of the DYO's that send email have their Close Only box checked so they evaluate once per bar.

Email is sent spaced at least 60 seconds apart. If email to send is pending, the message is inserted at the top of the email body and becomes the email Subject. Use the 'if # then Text Message' statement if you do not want the body to accumulate messages.

Move SELL Stop

An email is sent any time the Sell Stop is raised.

Message Text		Font	True	False	Panel
<input type="text" value="Move SELL Stop to [51] - CT [18] - HP [14] - WDD [13]"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sound		Trading System			Price
<input checked="" type="radio"/> Silent <input type="radio"/> Beep <input type="radio"/> Voice <input type="radio"/> WAV <input type="checkbox"/> Sound once per bar		<input type="text" value="Close"/>			
<input type="text" value="WAV file"/>		<input type="button" value="Browse"/>			
Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	
Branching	0	= if # then do next row			
<input type="button" value="←"/> <input type="button" value="→"/> <input type="button" value="↔"/>		([LowerMoved]	ANC	Position is Short	
A	Branching	if ([LowerMoved] AND Position is Short) then do next row			<input type="checkbox"/>
B	Action	if [Send Email] then E-mail 'Move SELL Stop to [51] - CT [18] - HP [14] -			<input type="checkbox"/>

A – Do the next row when the Lower Stop is raised and the Position is Short.

B – Conditionally send the email. The message for the email comes from the Message Text edit box. Variables in the text are resolved to show their values.

- [51] is the Lower Stop value
- [18] is the Current Trade profit
- [14] is the Maximum Profit
- [13] is the Maximum Draw Down

Move BUY Stop

This DYO is similar to the Move Sell Stop DYO. It will send an email when the Buy Stop is lowered and the Position is Long.

Message Text		Font	True	False	Panel
<input type="text" value="Move BUY Stop to [51] - CT [18] - HP [14] - WDD [13]"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sound		Trading System			Price
<input checked="" type="radio"/> Silent <input type="radio"/> Beep <input type="radio"/> Voice <input type="radio"/> WAV <input type="checkbox"/> Sound once per bar		<input type="text" value="Close"/>			
<input type="text" value="WAV file"/>		<input type="button" value="Browse"/>			
Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	
Branching	0	= if # then do next row			
<input type="button" value="←"/> <input type="button" value="→"/> <input type="button" value="↔"/>		([UpperMoved]	ANC	Position is Long	
A	Branching	if ([UpperMoved] AND Position is Long) then do next row			<input type="checkbox"/>
B	Action	if [Send Email] then E-mail 'Move BUY Stop to [51] - CT [18] - HP [14] -			<input type="checkbox"/>

Profit Target

This DYO will email a notification when the Profit Target objective has been achieved.

Message Text		Font	True	False	Panel
Profit Target Hit [201] - Last [\$C] - HP [14.2] - WDD [13]		Black	Green	Red	Yellow
Sound			Trading System		
<input checked="" type="radio"/> Silent <input type="radio"/> Beep <input type="radio"/> Voice <input type="radio"/> WAV <input type="checkbox"/> Sound once per bar			Price: <input type="text" value="Close"/> Quantity: <input type="text" value="1"/>		
WAV file		<input type="text" value=""/> <input type="button" value="Browse"/>			
Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Expression	0	= Profit Dollars	>=	[Profit Target]	0
		{			0
A	Expression	[0] := Profit Dollars >= [Profit Target]			<input type="checkbox"/>
B	Flag	[0] := [A] goes True			<input type="checkbox"/>
C	Action	if [Send Email] AND [B] then E-mail 'Profit Target Hit [201] - Last [\$C] - HP			<input type="checkbox"/>

A – This is a Boolean flag that is true when the current trade's profit reaches the Profit Target.

B – This is a flag that is only true when row B goes True.

C – Conditionally send the email when row B is True. The email's text is from the Message Text edit box.

[201] is the Profit Target value

[\$C] is the bar's Close value

]14.2] is the Maximum Profit, formatted for 2 decimals.

[13] is the maximum Draw Down

BUY/SELL Entry

This DYO will send an email when a new position is taken.

A	Expression	[0] := Long Entry Price	<input type="checkbox"/>		
B	Action	if Position goes Long AND [Send Email] then E-mail 'BUY @ [A]. Stop @	<input checked="" type="checkbox"/>	Study Value	BUY @ [A]. Stop @
C	Expression	[0] := Short Entry Price	<input type="checkbox"/>		
D	Action	if Position goes Short AND [Send Email] then E-mail 'SELL @ [C]. Stop @	<input checked="" type="checkbox"/>	Study Value	SELL @ [C]. Stop @

A & B – Send an email when a Long Position is taken. The message text is located in the row B label.

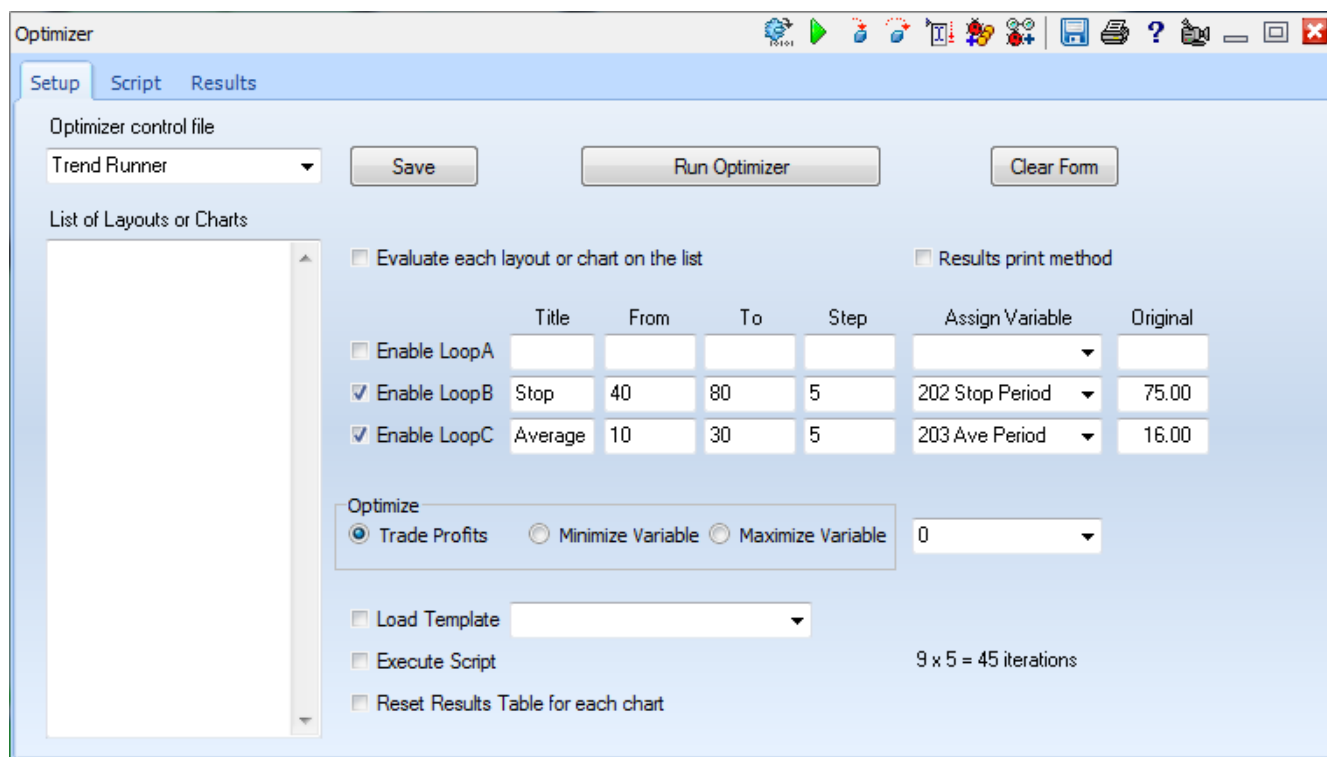
C & D – Send an email when a Short Position is taken. The message text is located in the row D label.

Optimizer

Now that the Trend Runner system is designed and looks promising, it is then time to employ the Optimizer to rapidly generate comparative results.

Row A of the Study Settings DYO performs an important purpose for the Optimizer. When the Optimizer is running, it is the Optimizer that is setting the values in the row B and C variables, and thus setting values by the DYO needs to be avoided. Therefore, the Optimizer flag is tested and when it is True the Optimizer is running. When True the execution of row B and row C is skipped.

To run the Optimizer, click on the Optimizer button on the Setup ribbon.



Loop A is not used in this example, so the Enable LoopA check box is unchecked.

Loop B will iterate the stop look back period in variable 202. It will be evaluated from 40 to 80 in increments of 5, such as 40, 45, 50 and so forth.

Loop C will iterate the moving average parameter in variable 203 between 10 and 30 in increments of 5.

The objective is to maximize Trade Profits and so that option is checked in the Optimize section.

Because the chart to optimize is already open and dressed with the trading system, there is no need to load any template or iterate through a list of layouts or charts.

Now click the Run Optimizer button, and the Results table will be populated.

	Chart		Stop	Average	Win #	Loss #	Total #	% Win	\$ Win	\$ Loss	Profit	Ratio	Average
23	EUR_USD.R100	0.0000	55.00	30.00	13	23	36	36	9653	-9190	463	1.0504	12.86
25	EUR_USD.R100	0.0000	60.00	10.0000	13	23	36	36	9402	-9082	320	1.0352	8.8881
26	EUR_USD.R100	0.0000	60.00	15.00	13	22	35	37	9851	-8420	1431	1.1699	40.89
27	EUR_USD.R100	0.0000	60.00	20.00	13	23	36	36	9406	-8946	460	1.0514	12.78
28	EUR_USD.R100	0.0000	60.00	25.00	13	23	36	36	9551	-8946	605	1.0676	16.80
29	EUR_USD.R100	0.0000	60.00	30.00	13	23	36	36	9551	-8946	605	1.0676	16.80
31	EUR_USD.R100	0.0000	65.00	10.0000	13	16	29	45	9838	-6172	3666	1.5940	126.41
32	EUR_USD.R100	0.0000	65.00	15.00	13	16	29	45	9926	-6172	3754	1.6082	129.45
33	EUR_USD.R100	0.0000	65.00	20.00	12	18	30	40	9694	-7248	2446	1.3375	81.53
34	EUR_USD.R100	0.0000	65.00	25.00	12	18	30	40	9839	-7248	2591	1.3575	86.37
35	EUR_USD.R100	0.0000	65.00	30.00	12	18	30	40	9839	-7248	2591	1.3575	86.37
37	EUR_USD.R100	0.0000	70.00	10.0000	11	15	26	42	9707	-5818	3889	1.6684	149.58
38	EUR_USD.R100	0.0000	70.00	15.00	11	15	26	42	9934	-5818	4116	1.7075	158.31
39	EUR_USD.R100	0.0000	70.00	20.00	11	15	26	42	9906	-5818	4088	1.7026	157.23
40	EUR_USD.R100	0.0000	70.00	25.00	11	15	26	42	9906	-5818	4088	1.7026	157.23
41	EUR_USD.R100	0.0000	70.00	30.00	11	15	26	42	9906	-5818	4088	1.7026	157.23
43	EUR_USD.R100	0.0000	75.00	10.0000	10	12	22	45	10314	-3462	6852	2.9792	311.46
44	EUR_USD.R100	0.0000	75.00	15.00	10	12	22	45	9578	-4234	5344	2.2622	242.91
45	EUR_USD.R100	0.0000	75.00	20.00	10	12	22	45	9578	-4234	5344	2.2622	242.91
46	EUR_USD.R100	0.0000	75.00	25.00	10	12	22	45	9578	-4234	5344	2.2622	242.91
47	EUR_USD.R100	0.0000	75.00	30.00	10	12	22	45	9578	-4234	5344	2.2622	242.91
49	EUR_USD.R100	0.0000	80.00	10.0000	8	13	21	38	9871	-3962	5909	2.4914	281.38
50	EUR_USD.R100	0.0000	80.00	15.00	8	12	20	40	9968	-3876	6092	2.5717	304.60
51	EUR_USD.R100	0.0000	80.00	20.00	8	12	20	40	9968	-3876	6092	2.5717	304.60
52	EUR_USD.R100	0.0000	80.00	25.00	8	12	20	40	9968	-3876	6092	2.5717	304.60
53	EUR_USD.R100	0.0000	80.00	30.00	8	12	20	40	9968	-3876	6092	2.5717	304.60

The highlighted values are the ones that were the best for that section. The table can be scrolled to show all results, and various settings considered. The settings that yielded the Maximum Profit will be shown with a highlight in the outer most loop, and that is row 43 in the example. The best settings of 75 and 10 are then posted back on the Setup tab.\

These results are also written to a text file in the C:\Ensign10\Optimizer folder using the chart's filename.

Trade Report

The optimized settings added about a thousand dollars to the Account Balance. Here is the trade history using the optimized settings. Click the Trade Report sub-menu of the Account button on the Main ribbon to show this form.

Trades		New	Edit								
Account Balance	\$	6,852.00	Winners	11	50	\$	10,314.00	Average Win	\$	937.64	
Market Value	\$	0.00	Losers	11	50	\$	-3,462.00	Average Loss	\$	-314.73	
Today's Change	\$	-116.99	Total	22		\$	6,852.00	Average Trade	\$	311.45	
Symbol	Pos	Quantity	EntryDate	EntryPrice	ExitPrice	Profit/Loss	Balance				
1 EUR/USD	S	1	02-09-12	1.32845	1.32845	\$ 0.00	\$ 0.00				
2 EUR/USD	L	1	02-09-12	1.33018	1.32715	\$ -303.00	\$ -303.00				
3 EUR/USD	S	1	02-09-12	1.32742	1.33058	\$ -316.00	\$ -619.00				
4 EUR/USD	L	1	02-09-12	1.32970	1.32600	\$ -370.00	\$ -989.00				
5 EUR/USD	S	1	02-10-12	1.32496	1.32613	\$ -117.00	\$ -1,106.00				
6 EUR/USD	L	1	02-13-12	1.32714	1.32121	\$ -593.00	\$ -1,699.00				
7 EUR/USD	S	1	02-13-12	1.32000	1.31796	\$ 204.00	\$ -1,495.00				
8 EUR/USD	S	1	02-15-12	1.31594	1.30340	\$ 1,254.00	\$ -241.00				
9 EUR/USD	L	1	02-16-12	1.30275	1.32345	\$ 2,070.00	\$ 1,829.00				
10 EUR/USD	S	1	02-21-12	1.32248	1.32541	\$ -293.00	\$ 1,536.00				
11 EUR/USD	L	1	02-23-12	1.32631	1.34164	\$ 1,533.00	\$ 3,069.00				
12 EUR/USD	S	1	02-27-12	1.34093	1.34388	\$ -295.00	\$ 2,774.00				
13 EUR/USD	L	1	02-28-12	1.34462	1.34216	\$ -246.00	\$ 2,528.00				
14 EUR/USD	S	1	02-29-12	1.34312	1.32221	\$ 2,091.00	\$ 4,619.00				
15 EUR/USD	L	1	03-05-12	1.32171	1.31754	\$ -417.00	\$ 4,202.00				
16 EUR/USD	S	1	03-06-12	1.31635	1.31631	\$ 4.00	\$ 4,206.00				
17 EUR/USD	L	1	03-07-12	1.31708	1.32270	\$ 562.00	\$ 4,768.00				
18 EUR/USD	S	1	03-09-12	1.32191	1.31436	\$ 755.00	\$ 5,523.00				
19 EUR/USD	L	1	03-12-12	1.31352	1.30926	\$ -426.00	\$ 5,097.00				
20 EUR/USD	S	1	03-13-12	1.30832	1.30571	\$ 261.00	\$ 5,358.00				
21 EUR/USD	S	1	03-15-12	1.30492	1.30578	\$ -86.00	\$ 5,272.00				
22 EUR/USD	L	1	03-15-12	1.30523	1.32103	\$ 1,580.00	\$ 6,852.00				

The system was designed for and run on a EUR/USD 100 Range chart with data from Feb 9th through March 21st, 2012. The optimized system made 22 trades in that period of time as shown in the trade report.

Parameter Automation

The results from the Optimizer are saved in a text file in the \Optimizer\ folder using the chart's filename.

The C:\Ensign10\Optimizer\EUR_USD.R100.txt file has this content written by the Optimizer.

0.0000, 75.00, 10.0000

A DYO can read these results and automatically adopt the optimized settings. The Study Settings DYO in this example could be changed to the following to set its variables with the Optimizer's results.

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Function	202 Stop Period	= Optimizer Field(#2), Defau	2		0
		{ 75			0
A	Branching	if Optimizer Running OR (Initialization Flag = False) then go ahead 4			
B	Function	[Stop Period] := Optimizer Field(2), Default (75)			
C	Function	[Ave Period] := Optimizer Field(3), Default (15)			

A – Row B and C are skipped when the Optimizer is running, or when the Initialization Flag is false. The Initialization Flag is True for the first execution call when the chart is evaluated. Once the Stop Period and Ave Period are initialized, it is a speed improvement to skip them for the rest of the chart's evaluation.

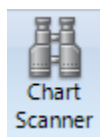
B – The Stop Period variable will be set to the Optimizer's Loop B result which is the 2nd field in the Optimizer results file. If the file does not exist, the Default value of 75 is assigned.

C – The Ave Period variable will be set to the Loop C result which is the 3rd field in the file. If the file does not exist, the Default value of 15 is assigned.

Template

This example trade system can be downloaded from the Ensign web site using the package feature. The package is named Template-TrendRunner and is found in Ensign's Official Folder. The template will be installed in the Template2 folder.

Chart Scanner



The Chart Scanner form is used to select a symbol list to be scanned. A chart will be opened for each symbol on the list in the selected time frame. A template can be optionally applied to the chart.

A criteria can be used to include the symbol on the report that is designed on the form. And the results can be viewed and printed.

Click the Chart Scanner button found on the Setup ribbon to display this form.

Enter a name in the Scanner control file combo box. After the properties have been filled in, or after changes are made, click the Save button to create or update the control file.

Symbol List

Use the 1st frame to Pick a List to Scan. The symbol list could be the contents of a Custom quote page, or it could be a symbol list for one of the Vendor feeds. Use the combo list boxes to make a page selection.

Enter in the Chart Time Fame edit box the chart time frame suffix, such as D for a daily chart, or 5 for a 5-minute chart. Multiple time frames can be entered separated by commas. Example: 5,60,D will open 3 chart time frames for each symbol.

Select a Template from the Template combo box if each chart is to be dressed with a template. Typically a template would be applied and the template would contain DYOs that define a Boolean variable that is True for when the chart is to be included in the report. The DY0 logic could be testing for a gap open, or for bar and study relationships, etc.

Criteria

Use the 2nd frame to select a Criteria for Adding to the Report. Check the 1st box if every symbol is to be added to the report. If a symbol is to be conditionally added to the report, then the test condition could be when a Variable is true, or when an ESPL function call returns a True result. If the logic for inclusion is determined by the template, then check the Variable is True box and use the drop down list to select the Variable that has the Boolean result.

Check the Function is True box when the logic is determined by an ESPL script. Enter in the edit box the function name to call in the script, and write the function on the Script tab of this form. The function you write needs to return a Boolean result.

Report

Use the 3rd frame to Design the Report. Each entry on the list creates a column in the report. A typical entry would contain both a column title and a reference tag separated by a space character and equal sign. Click on the [?] button in this frame for a list of the reference tags in the DYO manual. [Appendix – Reference Tags](#)

In this example, the 1st column of the report will be the Symbol for the chart. [\$S] is the reference tag that prints the chart symbol in the report. Or use [\$U] to show the symbol with its timeframe.

The 2nd column contains the daily High from the quote page record. [^H] prints the daily high.

The 3rd column contains the daily Low from the quote page record. The example prints on the report the text string which follows the 1st space = characters. The text is 'Low = ' and the value for the [^L] reference tag. This example is showing that text and tags can be mixed.

The 4th column prints the value from a Variable that is named [Bollinger Upper]. The variable was set by the DYO or study in the template. This example shows that Variables can be referenced by name.

The 5th and 6th columns show additional study values from Variables, and the variables are referenced by number. These variables could have been assigned directly on the study's property form.

The 7th column prints the results of a Variable that holds a Boolean result from a study test. The report will show 'True' or 'False' for a Boolean result.

The last entry on the list is just the column title of 'BarCount'. The report content for this column will come from the execution of a Script. Check the Execute Script box when a script is to be run to post results to the report. An optional name can be entered for a specific Procedure to call in the script. In this example, the entire script will be run because the Execute Script edit box is blank.

Design the Report

Enter: Column Title Text = Reference Tag ?

Symbol = [\$S]
High = [^H]
Low = Low = [^L]
Upper Bol = [Bollinger Upper]
Middle = [11]
Lower Bol = [12]
MACD>AVE = [28]
BarCount

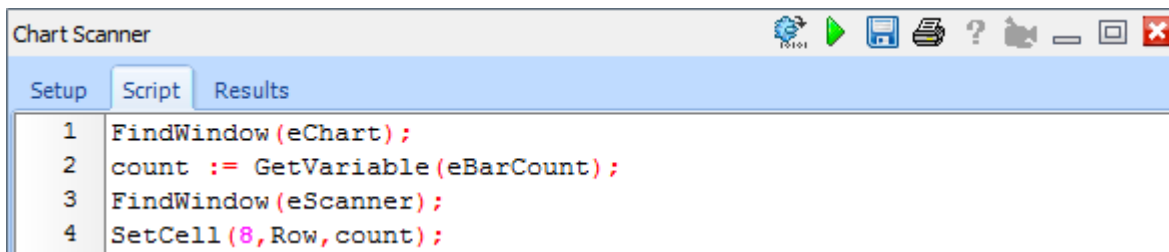
Execute Script

Scanner print method

Script

Scripts provide flexibility to design more complex logic and control. However the majority of Chart Scans will not utilize a script. This example shows a script to introduce a typical use.

Click the Script tab on the Scanner form to enter a script in the editor.



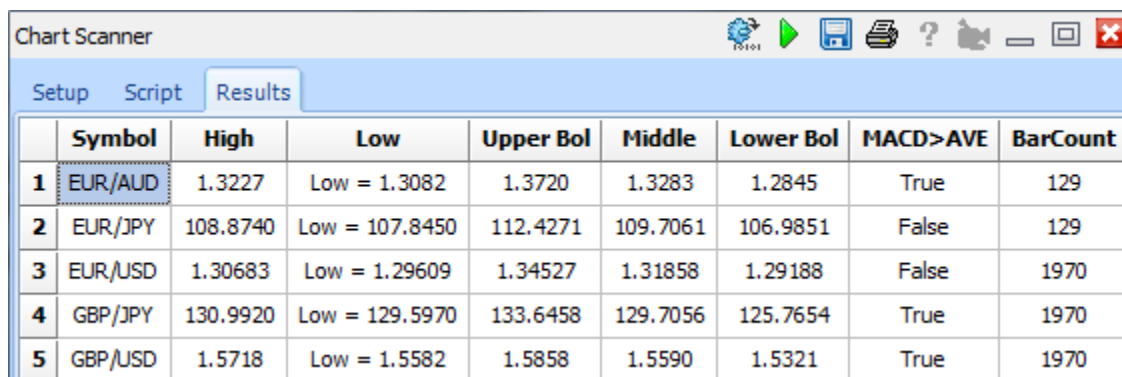
```
1 FindWindow (eChart) ;
2 count := GetVariable (eBarCount) ;
3 FindWindow (eScanner) ;
4 SetCell (8,Row, count) ;
```

Lines 1 and 2 locate the chart being scanned, and reads the chart's Bar Count variable. This returned value is assigned to a variable named 'count'.

Lines 3 and 4 then find the Scanner form, and writes the 'count' value into the 8th column. The 2nd parameter of the SetCell statement is the cell row. The example uses a global variable named 'Row' that contains the grid sheet's row number that the program is writing to. Lines 3 and 4 are typical of the 2 statements needed for writing content to the results table.

Results

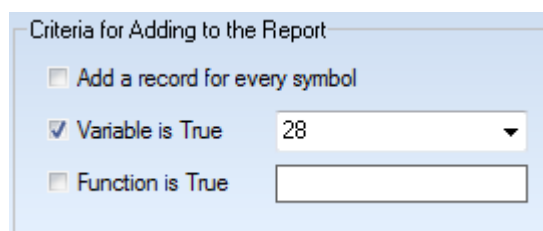
Click the Run Scanner button on the Setup tab to scan the charts in the symbol list and generate a report based on the criteria. The report will be shown on the Results tab.



	Symbol	High	Low	Upper Bol	Middle	Lower Bol	MACD>AVE	BarCount
1	EUR/AUD	1.3227	Low = 1.3082	1.3720	1.3283	1.2845	True	129
2	EUR/JPY	108.8740	Low = 107.8450	112.4271	109.7061	106.9851	False	129
3	EUR/USD	1.30683	Low = 1.29609	1.34527	1.31858	1.29188	False	1970
4	GBP/JPY	130.9920	Low = 129.5970	133.6458	129.7056	125.7654	True	1970
5	GBP/USD	1.5718	Low = 1.5582	1.5858	1.5590	1.5321	True	1970

The example scanned a list of 7 charts, and posted an entry for each symbol. The daily high and low are shown in their columns. Study values and Boolean flags are shown in their columns. And the Script posted the Bar Count from each chart in the 8th column.

Variable [28] is the Boolean flag shown in the MACD>AVE column. The criteria could have been changed to only report those charts with a True value by checking the Variable is True box and have 28 selected in the Variable's drop down list.

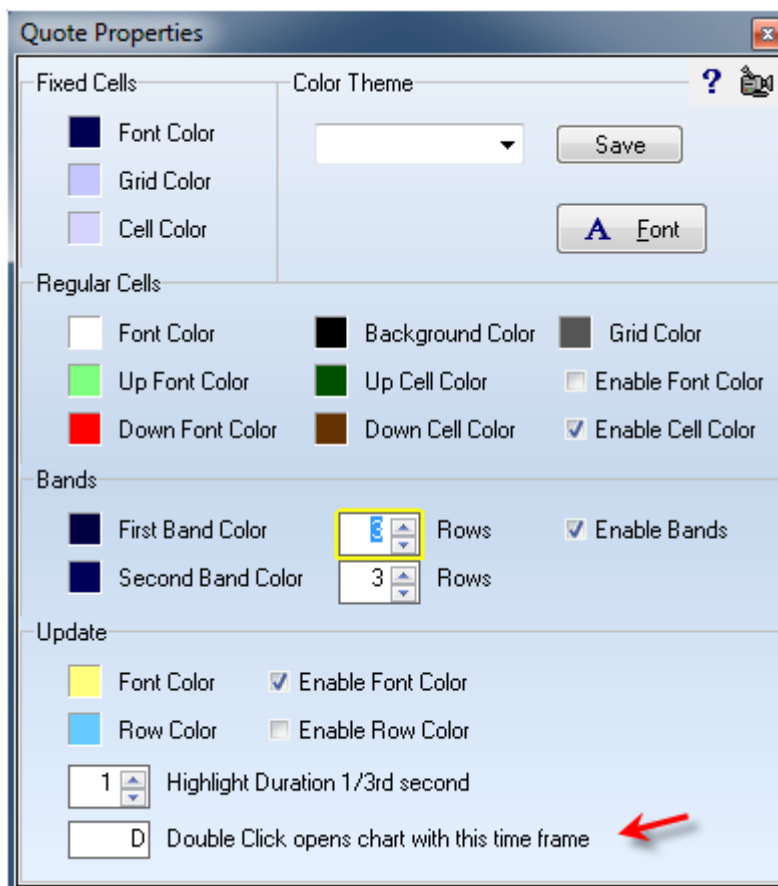


Criteria for Adding to the Report

- Add a record for every symbol
- Variable is True
- Function is True

Open a Chart

When one of the columns has a title of Symbol, clicking on a row in the Results table will open a chart for the symbol. If the [\$U] token was used to populate the symbol cell, the chart's time frame is included in the cell's text and will be used to open that chart. If the [\$S] token was used, the cell has just the symbol, and a default time frame read from the Quote page's property form will be used.



Sorting

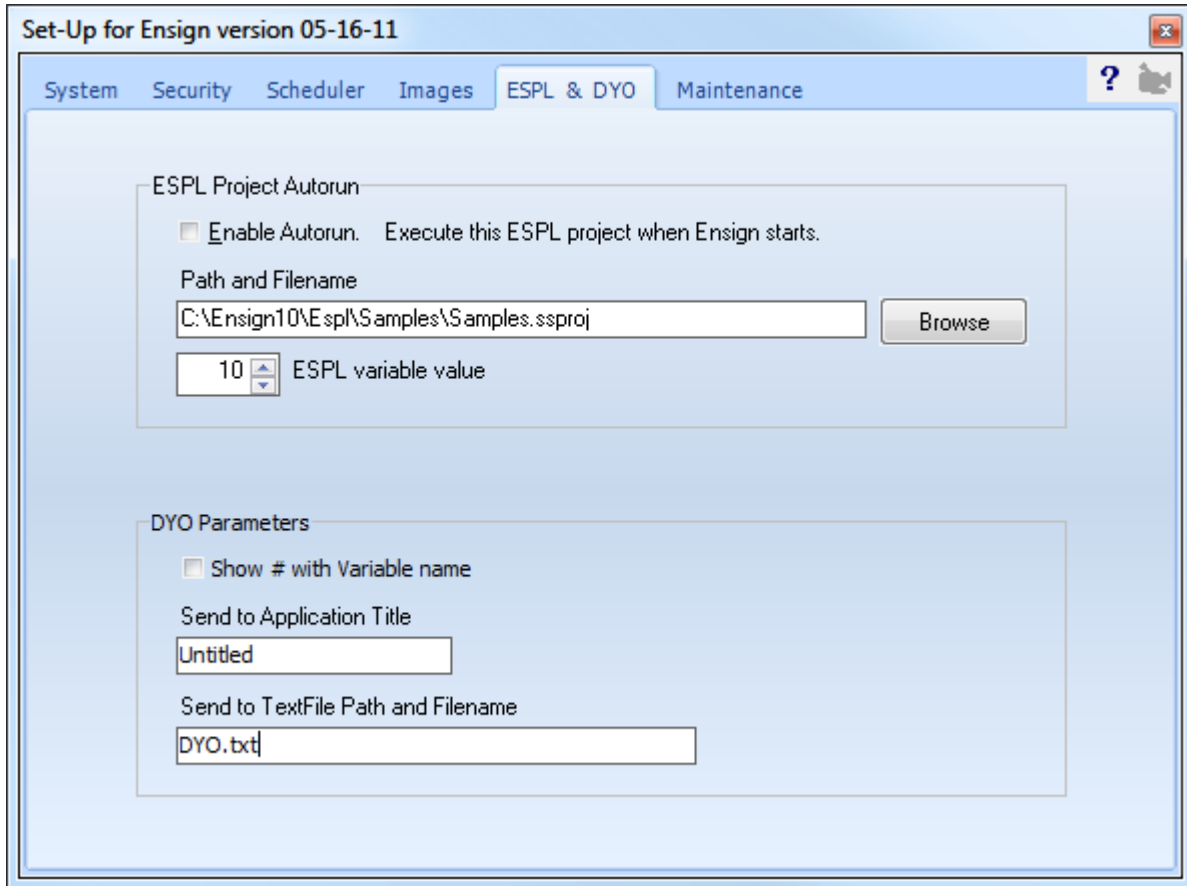
Click on a column title to sort the report based on the column values. Click the same column title again to change the sort direction. The sort direction will toggle between ascending and descending order.

Save to File

The pop-up menu for the scanner has menu selections to save the results to an ASCII file. The filename used will be the name of the scanner control file. The file will be saved in the C:\Ensign10\Ascii folder.

DYO Setup

Click menu Setup | System | ESPL & DYO to display the following form where three DYO parameters are configured.



Show # with Variable name

Check this option to show the index from the Variables array along with the variable name you have assigned. This will change the DYO script display, as in this example.

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show	Marker	Color
Chart Value	61 High	= #2 Minute High		60	0	<input checked="" type="checkbox"/>	[#]	[Green]
		(0)	
A	Chart Value	[61 High] := 60 Minute High				<input checked="" type="checkbox"/>	[#]	[Green] Study Value
B	Chart Value	[60 Low] := 60 Minute Low				<input type="checkbox"/>		

Send to Application Title

The Action | Send statements send a string via a Windows message to another application. Enter the title of the application on the Setup | System | ESPL form.

The following Delphi code shows how to receive and decode the Windows message.

```
FUNCTION TForm1.OnAppHook(var MyMessage: TMessage): boolean;
  var s: string;
  begin
    Result := false;
    case MyMessage.Msg of
      WM_COPYDATA: begin
        s := lpctr(pCDS(MyMessage.lParam)^.lpData); //string
        Result := true;
      end;
    end;
  end;
```

Appendix

Expression Data Values

Selection	Description
Close, High, Low, Open, Volume	Bar values
Net, abs(Net)	Net = Close – Prior Close, Absolute(Net)
Tick Count	Number of ticks in an intra-day or tick based bar.
Open Interest	Applies only to daily futures charts
Ask Volume, Bid Volume	Bar values. Volume = Ask Volume + Bid Volume
True, False	Boolean flags: True has a value of 1, False has a value of 0
Prior selections	Values from the prior bar
Range	A bar's High - Low
True Range	True Range High – True Range Low
True Range High, True Range Low	Highest(High, Prior Close), Lowest (Low, Prior Close)
Amount to complete bar	Values for portion remaining
Percent to complete bar	Value as a percent of portion remaining
Ask Ratio, Bid Ratio	Ask Volume / Volume, Bid Volume / Volume
Buy Pressure, Sell Pressure	Proprietary Ensign formula
Buy Ratio, Sell Ratio	Buy Pressure / Volume, Sell Pressure / Volume
Formula involving C, H, L, O	C = Close, H = High, L = Low, O = Open
Today High, Low, Open	Daily bar values considering both sessions
Today True Range High	Highest(Today High, Yesterday Close)
Today True Range Low	Lowest(Today Low, Yesterday Close)
Yesterday Close, High, Low, Open	Daily bar values for yesterday
Session 1 Close, High, Low, Open	A bar's 1 st session values (Day Session)
Session 2 Close, High, Low	A bar's 2 nd session values (Globex Session)
Date, Time	A bar's time stamp
Delta Time	Number of seconds between time stamps
Time Frame Type	0 = Tick, 1 = Range, 2 = Volume, 3 = Renko, 4 = Point & Figure, 5 = Second, 6 = Minute, 7 = Day, 8 = Week, 9 = Month, 10 = Quarter, 11 = Year
Time Frame Value	Value for a chart time frame, ex 5 for 5-minute bar
Second, Minute of the Day, Year	Values for a bar's time stamp from midnight or from Jan 1 st .
Minute, Hour, Day, Week, Month, Quarter, Year	Values for a bar's time stamp
Day of Week	Bar's time stamp: 1 = Sunday, 7= Saturday
Second, Minute of the Day, Year (Now)	Values for Now from midnight or from Jan 1 st .
Now as TDateTime	Integer part is Days from 1900. Fraction is Time. 0.5 is noon.
Minimum Tick Size	Smallest tick interval. Example: 0.25 for ES futures.

Tick Size Property	Tick value set on the chart's property form, on Symbol tab.
\$ / Tick Property	Value from chart property form, on Symbol tab.
Leverage Multiplier	Value from chart property form. Combination Tick and \$/Tick
Scale High, Low, Midpoint, Range, Grid Interval	Chart scale values
Scale Factor	Number of decimals, -2 = 1/8, -3 = 1/32
Candle Wick, Candle Tail	High – Highest(O,C), Lowest(O,C) - Low
Sub-window Scale High, Low, Midpoint, Range	Returns scale values for the DY0's subwindow location
Bar Count	Number of bars in the chart's data set
Bar Index	The index of a bar between 1 and Bar Count
Index is Last Bar	True when the Last visible bar is being evaluated
Visible Bar Count	The number returned is the number of visible bars
Count from 1st Visible Bar	The number returned is bar index – index of 1 st visible bar + 1
Index(selection)	The index of the bar which sets the selected price
In Session 1, 2	True when time stamp is in 1 st session (Day) or 2 nd session (Globex)
Color, Color Normal, Bullish, Bearish	Colors used by the chart background, bars, grids
Entry Price, Exit Price	Trade prices for last trade
Long Entry Price, Short Entry Price	Entry price for last trade
Buy Stop, Buy Limit, Sell Stop, Sell Limit	Set these values and Ensign will execute trades when touched
Commission	Amount to deduct for a round trip per trade quantity
Position Size, abs(Position Size)	Trading system position. Long > 0, Short < 0
Position Flat, Position Long, Position Short	Boolean flags based on the trade Position
Profit Points, Profit Percent, Profit Dollars	Trading system result for last trade
Total Win, Loss, Profit Dollars	Trading system results in dollars
Total Win, Loss Trades	Trading system results in number of trades
Sunday through Friday Open and Close times	Returns session times such as 930 and 1600
Initialization Flag	True when the Bar Index = 2. Useful for when to reset variables
Resize Event Flag	True when the chart resized and all studies are recalculating
Optimizer Running	True when the Optimizer is executing
Stack Tab Index	Returns the tab position on a stack. 0 means not on a stack.
Layer Number	Returns the layer the chart is on, ie. 1 through 9
Cursor Price	Price or study value the mouse cursor is vertically aligned with
Cursor Index	The bar index the mouse cursor is horizontally aligned with
Mouse X, Mouse Y	Return the mouse coordinates with (0,0) being the top left corner
Mouse Down	True if the left mouse button is currently held down
Mouse Click, Mouse Double Click	Flags that are set True by mouse actions, and cleared by reading
Mouse Click X, Mouse Click Y	Return the coordinates where the left mouse button was clicked.
Shift Pressed, Ctrl Pressed, Alt Pressed	Flags that are set True by pressing these keys, and cleared by reading
Key Down	Return the Key value for the last key pressed. Cleared by reading
Very High Number, Very Low Number	Returns 4,294,900,000 and -4,294,900,000

High Alert, Low Alert

Values set manually by CTRL-A keys, or on the Alert form.

Key Pressed

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show Marker	Color
Action	0	= if ## then Speak(#2)		-1	0	<input type="checkbox"/>	
		{ Shift Pressed			0		You pressed the Sh
A	Action	if (Shift Pressed) then Speak(-1)				<input type="checkbox"/>	You pressed the
B	Action	if (Ctrl Pressed) then Speak(-1)				<input type="checkbox"/>	You pressed the
C	Action	if (Alt Pressed) then Speak(-1)				<input type="checkbox"/>	You pressed the Alt

In this example, the label text on each line which is not fully shown is like: 'You pressed the Shift key'. The text is spoken as a result of the Shift, Ctrl or Alt key being pressed.

Operators

Math Operators	Description
+, -, *, /, div	Add, subtract, multiply, divide, integer division
^	Power, ex: $5^2 = 25$, $5^3 = 125$
* -, / -	Multiply or divide a negative value
mod	Modulo is the remainder from a division returned as an integer
Boolean Operators	Description
AND, AND Not, OR, XOR, Not	Boolean logic
>=, >	Greater than or equal to, greater than
=, <>	Equal to, not equal to
<, <=	Less than, less than or equal to
X>, X<, X<>	Cross above, cross below, cross above or below

Function Category

Function Name	Description
round(#)	# is rounded to the nearest whole number.
trunc(#)	Truncates # to an integer.
frac(#)	Returns the fractional part of #.
abs(#)	Returns an absolute value, ie. a positive value.
sqr(#)	Returns the square of a number.
sqrt(#)	Returns the square root of a number.
exp(#)	Returns e raised to the power of #
ln(#)	Returns the natural logarithm of #, $\ln(e) = 1$
log10(#)	Returns the log base 10 of #
V(#2)	Return a variable's value where the index is field #2's value.
Sum(#2, [#])	Sums the #2 values in the set of size [#]
Simple Average(#2, [#])	Simple Average, #2 data source, [#] parameter
Exponential Ave(#2, [#])	Exponential Average, #2 data source, [#] parameter
Smoothed Ave(#2, [#])	Smoothed Average, #2 data source, [#] parameter
Weighted Ave(#2, [#])	Weighted Average, #2 data source, [#] parameter
Momentum(#2, [#])	Momentum, #2 data source, [#] parameter
Stochastic(#2, [#])	Stochastic formula, #2 data source, [#] parameter
Standard Dev(#2, [#])	Standard Deviation, #2 data source, [#] parameter
Hist. Volatility(#2, [#])	Historical Volatility, #2 data source, [#] parameter
Linear Regression(#2, [#])	Linear Regression, #2 data source, [#] parameter

Linear Reg. Slope(#2, [#])	Linear Regression Slope, #2 data source, [#] parameter
R-Squared(#2, [#])	R-Squared formula, #2 data source, [#] parameter
Fib(#2 * (#3 - #4) + #4)	Fibonacci Price Levels formula
Sto(100*(#2 - #4) / (#3 - #4))	Stochastic formula
Sqrt(sqrt(#2) + sqrt(##))	Pythagorean theorem. Find hypotenuse from 2 sides.
Highest(#2, [#])	Returns the highest #2 value in the set of size [#]
Lowest(#2, [#])	Returns the lowest #2 value in the set of size [#]
Highest(#2, ##)	Returns the highest #2 value in the set of size ##
Lowest(#2, ##)	Returns the lowest #2 value in the set of size ##
Minimum(#2, ##)	Return the smaller of 2 values
Maximum(#2, ##)	Return the larger of 2 values
Minimum(#2, #3, #4)	Return the smallest of the 3 values
Maximum(#2, #3, #4)	Return the largest of the 3 values
Nearest ## multiple of #2	Round ## up or down to the nearest multiple of #2
Nearest(#3, #4) oper #2	Return #3 or #4 when compared to #2
Bars since True(#)	Current Index – prior Index where # is True
Bars since False(#)	Current Index – prior Index where # is False
Count True(#2, [#])	Count the number of non-zero values in a set of variables from #2.
Count False(#2, [#])	Count the number of zero values in a set of variables from #2.
Index of last True(#)	Recent Index where # was True
Index of last False(#)	Recent Index where # was False
Index of Highest(#2, [#])	Returns the Bar Index for the highest #2 value in the set of size [#]
Index of Lowest(#2, [#])	Returns the Bar Index for the lowest #2 value in the set of size [#]
Index(Body inside prior #2)	
Index(Body covers prior #2)	
Index(Range inside prior #2)	
Index(Range covers prior #2)	
Pixels(#)	Pixels vertically of # relative to Scale Low
ArcTan Degrees(#)	ArcTan(#) * 180 / pi
Pyrapoint Degrees(#2, ##)	Pyrapoint formula converts ## for #2 degree rotation
Protractor Chart(#)	Arctan(# in pixels / bar spacing) * 180 / pi
Protractor Subwindow(#)	Arctan(# in pixels / bar spacing) * 180 / pi
#2 as % of Sub-window Scale	Returns 100 * (#2 value – sub-window low) / sub-window range
IndexToX(#2)	Return the horizontal plot position for a bar's index.
XToIndex(#2)	Return a bar's index based on its horizontal plot position.
PriceToY(#2)	Return the vertical plot position for a price.
YToPrice(#2)	Return a price for a vertical plot position.
Optimizer File Field(#2)	Returns field #2 in file C:\Ensign10\Optimizer\chartfilename.txt
ASCII File Field(#2)	Returns field #2 in file C:\Ensign10\ASCII\chartfilename.txt

Flag Category

Flag Function	Description
AND(#2, [#])	AND a set of #2 values where [#] is the set size
OR(#2, [#])	OR a set of #2 values where [#] is the set size
#2 goes True	True when #2 changes from False to True
#2 goes False	True when #2 changes from True to False
#2 changes state	True when #2 changes state
#2 rising	True when #2 > prior #2
#2 falling	True when #2 < prior #2
#2 turns up	True when #2 slope goes positive
#2 turns down	True when #2 slope goes negative
#2 changes direction	True when #2 slope changes direction
1st Timestamp oper #2	True for 1 st bar with the operator relationship to time #2
1st tick of New bar	True for 1 st tick that starts a new bar
1st bar of New Minute	True for 1 st bar of a new minute period
1st bar of New Hour + #2 min	True for 1 st bar of a new hour period, offset #2
1st bar of New Day, or New Trade Day	True for 1 st bar of a new calendar date, or new trade day
1st bar of Session 1, 2	True for 1 st bar of 1 st session, or of 2 nd session
1st bar of Either Session	True for 1 st bar of a new session
1st bar of New Week, Month, Quarter, Year	True for 1 st bar of a new week, month, quarter, or year
#2 near #3 +/- #4	True when #2 >= #3 - #4 and #2 <= #3 + #4
#2 between #3 & #4	True when #2 >= #3 and #2 <= #4. Have #3 < #4
#2 between #3 & (#3 + #4)	True when #2 >= #3 and #2 <= #3 + #4
Index >= (Bar Count - #2)	True when Index >= Bar Count - #2
Sets High in 1st #2 min period	True when bar sets High in #2 period after open
Sets Low in 1st #2 min period	True when bar sets Low in #2 period after open
In #2 min after Session 1, 2 Open	True when time stamp in #2 period after 1 st or 2 nd open
In #2 min after Time #3	True when time stamp in #2 period following #3
In #2 min before Session 1, 2 End	True when time stamp near 1 st session close, or 2 nd session close
In #2 min before Time #3	True when time stamp near time #3
Date is Today, Date is Yesterday	True when time stamp is part of today's daily bar, or yesterday's bar
Key Reversal Pair	True when 2 bars form a Key Reversal Pair
Hammer(O,C >= #2%)	Both Open and Close >= #2 percent of range
Gravestone(O,C <= #2%)	Both Open and Close <= #2 percent of range
Spinner(O,C inner #2%)	Both Open and Close near the the center
Up Marubozu(O,C outer #2%)	Close > Open and both near range extremes
Dn Marubozu(O,C outer #2%)	Close < Open and both near range extremes
Ascending(#2, [#]), Descending(#2, [#])	All #2 values are ascending or descending in the set of size [#]

Action Category

Selection	Description
V := # ; plot V as study transfer	Retain values through a study recalculate
V := ##; plot V shifted #2 bars	Plot ## shifted left or right from normal position
V := # ; plot V in margin shifted [#] bars	Plot ## only in the right side margin shifted [#]
V := ##; if #2 oper 0 then plot V	Assign V, but conditionally plot based on #2 test
V := #2; if ## then plot V	Assign V, but conditionally plot based on ## test
inc(#2); V := ##	Increment #2, then assign V the result of ##
dec(#2); V := ##	Decrement #2, then assign V the result of ##
V(#2) := ##	Variable with index from #2 will be assigned the result of ##
if ## then V := #2	If the test is True then assign V the #2 value
if ## then V := #2 else V := 0	If the test is True then return #2 else return zero
if ## then V := #3 else V := #4	If the test is True then return #3 else return #4
if ## then V := (#3 - #4) else V := #2	If the test is True then return #3-#4 else return #2
if #2 oper 0 then V := ##	If the test is True then return the result of ##
if #2 oper 0 then V := #3 else V := #4	If the test is True then return #3 else return #4
if # then inc(V)	If the test is True then V := V + 1;
if # then dec(V)	If the test is True the V := V - 1;
if # then inc(V, #2)	If the test is True then V := V + #2; Note: #2 can be negative
if ## then inc(V) else inc(#2)	If True then V := V +1, else the variable in #2 is incremented.
Section Font Size := #2	Change the Section Font Size. Bold = negative
Label Font Size := #2	Change the Label Font Size. Bold = negative
Scale Mode := #2, High := #3, Low := #4	Set scale mode [0..5]. 2 is Use Range, provide Scale High and Low
if ## then marker in column #2	Test ## and plot a marker in column #2 when True
If ## then SetFocus(Change Layer = #2)	Bring a chart into view. If #2 is True the layer can change if needed
Row #2 := ##	Change the value on the row selected in #2
Row #2 Operator[1] := ##	Change the top row operator selected in #2
Row #2 Operator[2] := ##	Change the bottom row operator selected in #2
Row #2 Offset[1] := ##	Change the top row offset selected in #2
Row #2 Offset[2] := ##	Change the bottom row offset selected in #2
dec(Row #2 Offset[1])	Decrement the top row offset selected in #2
dec(Row #2 Offset[2])	Decrement the bottom row offset selected in #2
if ## then Action #2	See table of action selections for #2
Trade Quantity := #2	Set the trade system trade quantity to #2
if # then Stop and Reverse	Statements for implementing trading systems and back testing
if # then Buy or Reverse	
if # then Buy Market	
if ## then Buy Limit Price := #2	
if ## then Buy Stop Price := #2	

if # then Sell or Reverse	
if # then Sell Market	
if ## then Sell Limit Price := #2	See the Trade System example in this manual
if ## then Sell Stop Price := #2	
if # then Exit Position	
if # then Exit Long Position	
if # then Exit Short Position	
if # then Cancel Buy Limit	
if # then Cancel Buy Stop	
if # then Cancel Sell Limit	
if # then Cancel Sell Stop	
If # then Do Not Trade Now	Do not execute pending Stop and Limit orders. These trigger levels are not cleared. They just are not checked while the DoNotTradeNow flag is True.
Find Spreadsheet(Message)	Open Spreadsheet if not found. Label is the name
Cell(#3, #4) := Format(#2, [#])	Write #2 value to a spreadsheet cell[col,row], and marker and color [#] is number of decimals. 7 = hh:nn: 8 = hh:nn:ss 9 = mm-dd-yy 10 = ASCII(#2)
Cell(#3, #4) := Ascii(#2)	The ASCII character of #2 value is written to cell(col,row)
FontSize(#3, #4) := #2	Change the cell's font size, and write message, marker, cell color
FontBold(#3, #4) := #2 oper 0	Conditionally set the cell's font style to bold.
if #2 oper 0 then Fontcolor(#3, #4) :=	Change the cell's font color. Write message, and marker
if #2 oper 0 then Cell(#3, #4) := Message	Write Label or Message, Marker, and set cell color
if #2 oper 0 then ColWidth(#3) := #4	Change the column width. #4 is the width in pixels
if #2 oper 0 then RowHeight(#3) := #4	Change the row height. #4 is the height in pixels
if ## then Rowcolor(#2) := Color	Change the row's background color
Label := Cell(#3, #4)	Read a cell's text into the DYO row Label
V := Cell(#3, #4) oper #2	Assign Variable the expression Cell value operator #2
V := Cell(#3, #4) row move #2 cells	Fill the Variables array with data from a spreadsheet row
V := Cell(#3, #4) col move #2 cells	Fill the Variables array with data from a spreadsheet column
Cell(#3, #4) := V row fill #2 cells	Fill a spreadsheet row from the Variables array starting with V
Cell(#3, #4) := V col fill #2 cells	Fill a spreadsheet column from the Variables array starting with V
Spreadsheet.Recalculate	Force a recalculate so current values can be read
Spreadsheet.Clear	All cells in the spreadsheet will be cleared
Spreadsheet.Size(#3 cols x #4 rows)	Change the size of the spreadsheet
if ## then Speak(#2)	Text is the Variable #2 name which can be lengthy
if ## then Beep(#2)	#2 = [0..4] for 5 different beep sound selections
if # then Alert(Message) *** email option	Conditionally show an Alert panel with Message
if ## then Alert(#2) *** email option	Text is the Variable #2 name which can be lengthy for Alert
if # then Play Sound	Conditionally play the Sound selection
if # then Output(Message)	Write Label or Message to the Output Window

if ## then Output(#2)	Text is the Variable #2 name which can be lengthy
if # then Send(Message)	Send Label or Message to another application, or Ensign Exchange
if ## then Send([#], #2)	Send [#], symbol, #2 selection to an application
If # then SendKeys(Message)	Send Label or Message keys to an application
if # then AppendFile(Message)	Conditionally write Message to a TextFile
if ## then AppendFile(#2)	Text is the Variable #2 name which can be lengthy
if #2 then V := ReadFile(#3, #4)	Conditionally read a text file from column #3, row #4
if # then Clipboard(Message)	Conditionally copy Message to the Clipboard
if ## then Clipboard(#2)	Text is the Variable #2 name which can be lengthy
if # then E-mail Message	Conditionally e-mail the Message text
if # then E-mail Chart	Conditionally e-mail a chart image
if # then E-mail Layout	Conditionally e-mail the layout script file
if # then Text Message	Conditionally text (e-mail) the Message text. See Example.
if ## then Vars(#2, [#]) := 0	Assign a set of Variables the value 0
if ## then Vars(#2, [#]) := 1	Assign a set of Variables the value 1
if ## then Shift Right(#2, [#])	Shift rightward a set of Variables. #2 is like var [20]. Set size is [#]
if ## then Shift Left(#2, [#])	Shift leftward a set of Variables. Last value remains unchanged.
if ## then Rotate Right(#2, [#])	Rotate rightward a set of Variables. Example: Rotate([20], 5)
if ## then Rotate Left(#2, [#])	Rotate leftward a set of Variables. 1 st value moves to the end.
if ## then Sort Ascending(#2, [#])	Sort the set of Variables with non-zero values. Zeros will be at end.
if ## then Sort Descending(#2, [#])	Sort the set of Variables. Set size is [#]. Set start is the #2 variable.
if ## then Swap #3 and #4	Conditionally swap variable in #3 with #4
if #2 then Copy(#3,[#]) to #4	Copy a set of Variables. Example Copy([20], 5) to [40]
PriceFinder makes # True	Iterate prices until the expression is True
PriceFinder makes # False	Iterate prices until the expression is False
PriceFinder makes # Change	Iterate prices until the expression changes state
if #2 oper 0 then MoveTo(#3, #4)	Conditionally move the Pen without drawing. Sets a draw point.
if #2 oper 0 then LineTo (#3, #4)	Conditionally draw a line between 2 draw points.
if #2 oper 0 then HorzLine(#3, #4)	Conditionally draw a horizontal line of length #3, at price #4.
if #2 oper 0 then VertLine(#3, #4)	Conditionally draw a vertical line between price #3 and price #4.
if #2 oper 0 then Rectangle(#3, #4)	Conditionally draw a rectangle between 2 data points.
if #2 oper 0 then Circle(#3, #4)	Conditionally draw a circle of radius #3, at price #4.
if #2 oper 0 then Ellipse(#3, #4)	Conditionally draw an ellipse in the rectangle bounded by 2 points
if #2 oper 0 then Arc(#3, #4)	Conditionally draw an arc in the rectangle bounded by 2 points.
if #2 then Button[#].Image := #3 else #4	Change the button image on the 6 toolbar buttons. [#] is 1..6
Button[#].Visible := ##	Show or hide one of the 6 toolbar buttons. [#] is 1..6
object(#2).Set Point A(#3, #4), also B and C	Pass the object handle in #2, index in #3, and price in #4.
object(#2).Set Param 1(#3), also 2, 3, U/D, L/R	Pass the object handle in #2, and the value to set in #3.
object(#2).Set Field(#3, #4)	Set any property. See ESPL GetStudy for #3 selection numbers.

object(#2).Get Field(#3)	Read any property. See ESPL GetStudy for #3 selection numbers.
------------------------------	----------------------------------------------------------------

If Flag then - Statements of this type read the Flag expression, and conditionally perform the action, including plotting of an option marker in the Show section of the DYO row.

Price Finder

PriceFinder selections evaluate a Flag and return the price where the Flag will be True, False, or Change states. For example, PriceFinder can determine the price that would make two moving averages cross, or cause CCI to cross zero, or make a bar reach a channel band.

V := # ; plot V as study transfer

Use this statement when a value to be plotted was calculated on another chart and transferred in a Variable. This statement receives special treatment to preserve its values when the chart recalculates studies after a visit to a property form or when the chart is resized. The values previously passed need to be preserved because they will not be passed again by the other chart. Another user of this statement is to preserve values that are received in real-time from a quote page.

[read more » Studies: Transfer from Another Chart](#)

Row #2 Offset[1] :=

This statement is used to reference other bar data by changing the Offset property for a DYO row.

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset	Show
Action	0	= Row #2 Offset[1] := ##		[B]	0	<input type="checkbox"/>
		{ Cursor Index	-	Bar Count	0	
A	Action	Row [B] Offset[1] := (Cursor Index - Bar Count)				<input type="checkbox"/>
B	Expression	[0] := High[-29]			Study Value	<input checked="" type="checkbox"/> [B]

The example calculates the negative offset for the bar the mouse cursor is aligned with and sets Row B's Offset. Row B then reads the High of the bar the cursor points to and shows this value on the chart.

Action Statements involving Strings

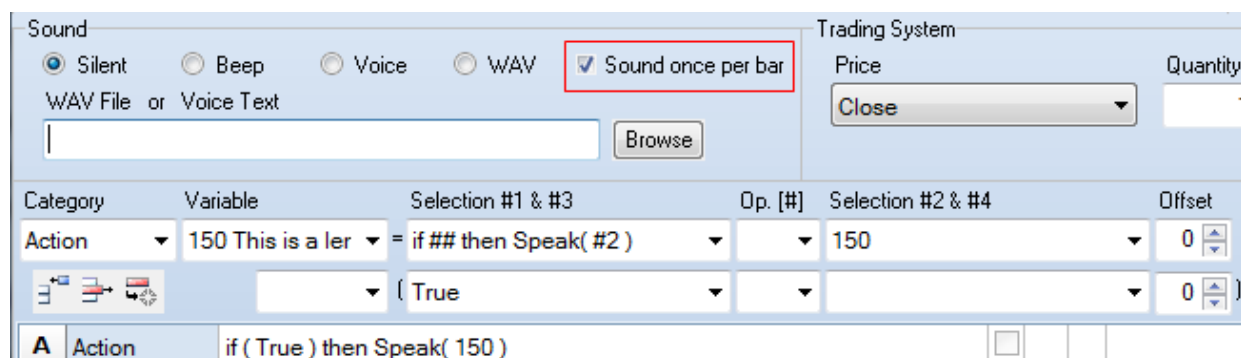
if ## then Speak(#2)

This statement can be used to implement voice alerts. The voice built into Ensign 10 comes from the Microsoft Windows API voice functions. The text that the voice will read is created according to the following priority sequence.

The #2 value is a Variable index, the text will be that variable's name. The variable name can be a lengthy message to be spoken. Enter the index of the variable, and not the variable.

If the variable name is blank, or the #2 value is less than 0 or greater than 999, then the text used will be the row's label. If the row's label string is blank, then the text will be the DYO form's Message Text.

The text may contain [reference tags](#) such as [\$S] and [\$C], which will be resolved before the text is sent to the speech engine. The speech is performed on a separate thread, so that program processing of the data feed is not paused.



The statement will also observe the state of the 'Sound once per bar' check box. When checked, the Speak will be restricted to being played once per bar.

The name for the [150] variable in this example is a lengthy string, such as:

'This is a lengthy message to be spoken. Have a great day!'

if # then Output(Message)

if ## then Output(#2)

The text string is written in the Output Window. These statements can be very useful to document the execution flow through the DYO's for debugging purposes.

if # then Clipboard(Message)

if ## then Clipboard(#2)

The text string is copied to the Windows Clipboard.

If # then Alert(Message)

If # then Alert(#2)

These statements can be used to show an alert message on the top row of the chart, right hand side. The colors for the font and panel are set on the DYO form.

The text for the alert comes from the following:

If the #2 value is a Variable index, and the text will be that variable's name. The variable name can be a lengthy message to be spoken.

If the variable name is blank, or the #2 value is less than 0 or greater than 999, then the text used will be the row's label. For Alert(Message), the text will be the row's label. If the row's label string is blank, then the text will be the DYO form's Message Text.

The text may contain [reference tags](#) such as [\$S] and [\$C], which will be resolved before showing the alert.

The Alert panel will display for a minimum of 10 seconds, and then auto hide, unless another alert triggers to extend the display period.

Email Alert

Check the Email check box to have the Alert emailed to the list of addresses configured on the Setup | Internet Services | Email form.

If # then Send(Message)

Message is a text string from the label on the DYO row. If the label is blank, then Message is the text from the Message Text edit box.

If ## then Send([#], #2)

The string that is sent is the Op. [#] field number, the chart symbol, and the value from the #2 field. The fields are separated by commas. Example: '5,EUR/USD,1.44950'

if # then SendKeys(Message)

This statement sends keyboard keystrokes to the application.

The following characters send the Alt, Ctrl, and Shift keys.

- & Alt key down. Holds the Alt key down while the next character is sent. This is used to access menu hot-keys. Menu hot-keys are not case sensitive. Example: &F is the same as pressing Alt-F. NOTE: Use {Alt} if you want a full keystroke of the Alt key.
- ^ Ctrl key down. Holds the Ctrl key down while the next character is sent. Example: ^C is the same as pressing Ctrl-C.
- ~ Shift key down. Holds the Shift key down while the next character is sent. Example: ~{Tab} is the same as pressing Shift-Tab.

The following tokens can be used to send the indicated keys.

{F1}	{F5}	{F9}	{Alt}	{Esc}	{Left}	{Return}
{F2}	{F6}	{F10}	{Backspace}	{End}	{PgDn}	{Right}
{F3}	{F7}	{F11}	{Del}	{Home}	{PgUp}	{Tab}
{F4}	{F8}	{F12}	{Down}	{Ins}	{PrtSc}	{Up}

The Action | AppendFile statements append strings to the named TextFile. TextFile is named on Setup | System | ESPL & DYO. The path will be in the C:\Ensign10\ folder. Enter the sub-folder (if any) and the name of the text file.

if # then AppendFile(Message)

Message is a text string from the label on the DYO row. If the label is blank, then Message is the text from the Message Text edit box. The message text is appended to the TextFile.

if ## then AppendFile(#2)

The #2 value is a Variable index, and the text will be that variable's name. The variable name can be a lengthy text string.

If the variable name is blank, or the #2 value is less than 0 or greater than 999, then the text used will be the row's label. If the row's label string is blank, then the text will be the DYO form's Message Text.

The text may contain [reference tags](#) such as [\$S] and [\$C], which will be resolved before the text appended.

Example: Variable 350 has the lengthy name of: 'Alert: Study Crossing Detected'

Entering 350 in Selection #2 will write this text to the TextFile when the ## expression evaluates to True.

if #2 then V := ReadFile(#3, #4)

The flag in #2 conditionally reads a value from a TextFile that is comma delimited. The field in the file will be in the #3 column on the #4 row. The columns and rows begin with a reference from 1.

Set-Up for Ensign version 04-01-13

System Security Scheduler Images **ESPL & DYO** Maintenance Buttons

ESPL Project Autorun

Enable Autorun. Execute this ESPL project when Ensign starts.

Path and Filename

C:\Ensign10\ESPL\Samples\Samples.ssproj

10 ESPL variable value

DYO Parameters

Show # with Variable name

Untitled

DYO.txt

ASCI\ES #F.txt

The TextFile for the AppendFile and the ReadFile statements are configured on the Setup | System | ESPL & DYO form. The root path of C:\Ensign10\ is automatically added to the path entries as a prefix.

If # then E-mail Message

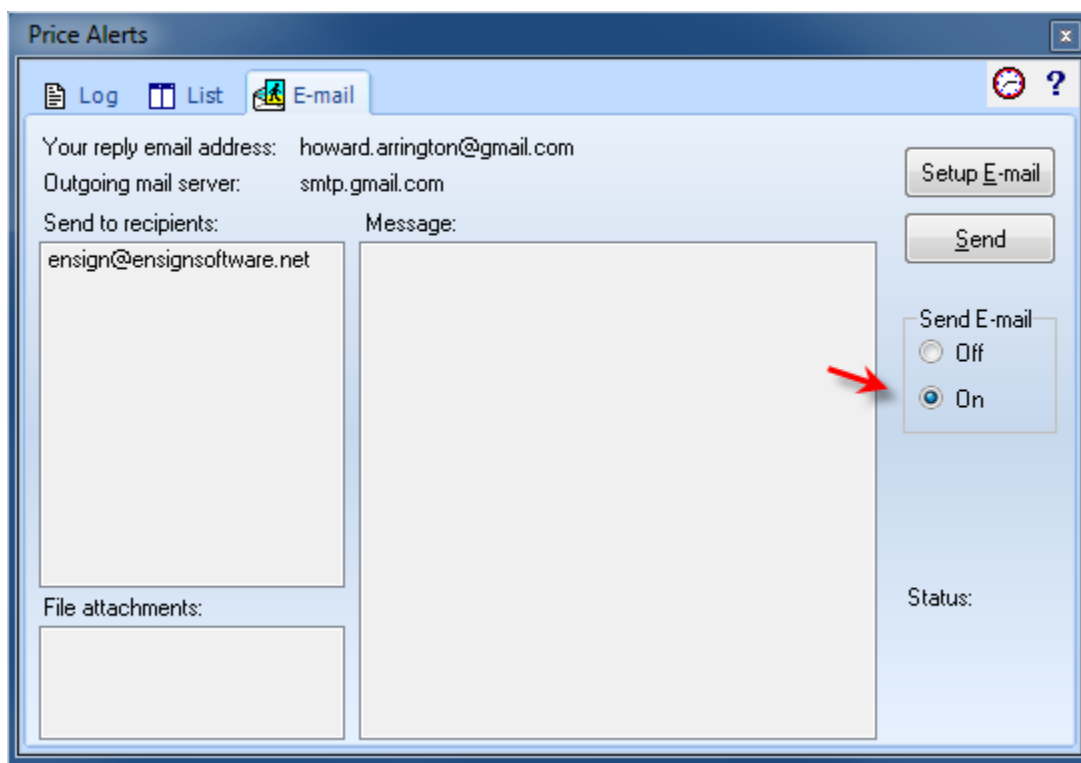
If # then Text Message

Message is a text string from the label on the DYO row. If the label is blank, then Message is the text from the Message Text edit box. The E-mail statement will automatically insert the chart's symbol as a prefix to the Message text.

Both statements send an email. The difference is the E-mail Message will accumulate Messages in the email body if an email is waiting to be sent. The Text Message will replace the email body and only send its Message text.

Email is sent no more frequently than every 60 seconds. An email or text message may have to wait for the 60 second timeout to expire before it is sent from the queue.

The email is sent to the list of addresses configured on the Setup | Internet Services | Email form. Also, the Send E-mail bullet on the Price Alerts form must be checked. This property globally enables or disables the sending of email by DYO's.



If # then Action #2

The action performed is selected by the value in Selection #2, according to the following list of available actions. Be very careful in using a DYO to programatically trigger an Action because some actions open other forms, and some initiate time consuming processes. Such actions would be inappropriate to execute with every tick update to the chart, or even with every bar completion. Warning: Unwise use of the Action feature can result in a severe burden to the CPU processing causing the system to become unresponsive.

Color Bar Actions

1 - Island Reversal	12 - Turning Points	23 - Full Moon
2 - Key Reversal Pair	13 - Small Trends	24 - Dunnigan
3 - Close Outer 10%	14 - Minor Trends	25 - Volume Increases
4 - Close Outer 25%	15 - Major Trends	26 - Open Outer 20%
5 - Gap	16 - Outside/Inside	27 - Narrow Range 7
6 - Gap Open	17 - Weekly High/Low	28 - Month by Month
7 - Net Change	18 - Minute by Minute	29 - Year by Year
8 - Close vs. Open	19 - Hour by Hour	30 - 4-Min by 4-Min
9 - Large/Small Range	20 - Day by Day	31 - Ask/Bid Volume
10 - Large/Small Volume	21 - New Month and Week	32 - Buy/Sell Pressure
11 - Trends	22 - Moon Phases	33 - Quarter by Quarter

Chart Actions

40 - Rebuild Visible Bars	60 - Toggle Calendar Days	190 - Compress Bar Spacing
50 - Refresh 1st quantity	98 - Toggle Show Scale	191 - Expand Bar Spacing
51 - Refresh 2nd quantity	99 - Align Charts	192 - Move Chart Up
52 - Refresh 3rd quantity		193 - Move Chart Down
53 - Refresh 4th quantity		194 - Zoom Out
54 - Refresh 5th quantity		195 - Zoom In
55 - Refresh 6th quantity		196 - Move Leftward
56 - Refresh 7th quantity		197 - Move Rightward
		198 - Default Scale
		200 - Repaint Chart
		201 - Recalculate Studies & Repaint

Keyboard Actions

100 - Time and Sales	111 - Reset Chart Defaults	180 - News Stories
102 - Snap Quote	112 - Trade Detail	181 - Toggle Bar Data Panel
103 - Print Chart	113 - Previous Chart	182 - Toggle Study Panel
104 - Print Data Set	114 - Next Chart	183 - Toggle Lines Panel
105 - Data to Clipboard	115 - Toggle Scroll	184 - Toggle Show Bars
106 - Image to Clipboard	116 - New Symbol	185 - Toggle Show Volume
107 - PNG Color Set	117 - Big Cross	186 - Toggle Show Studies
108 - Backspace Large Chart	118 - Save Chart Objects	187 - Toggle Show Lines
109 - Edit Data Set	119 - Close Chart	
110 - Delete Bar		

Study Category

Most Studies	Description
1 st , 2 nd , 3 rd line value	Study value
1 st , 2 nd , 3 rd line slope	Study value – prior study value
1 st line >= 2 nd line	Flag: 1 st line greater than or equal to 2 nd line
1 st line <= 2 nd line	Flag: 1 st line less than or equal to 2 nd line
1 st line X> 2 nd line	Flag: 1 st line cross above 2 nd line
1 st line X< 2 nd line	Flag: 1 st line cross below 2 nd line
1 st line X<> 2 nd line	Flag: 1 st line crosses 2 nd line
1 st line and 2 nd line rising	Flag: both line slopes are positive
1 st line and 2 nd line falling	Flag: both line slopes are negative
1 st , 2 nd , 3 rd line rising	Flag: study value => prior study value
1 st , 2 nd , 3 rd line falling	Flag: study value <= prior study value
1 st , 2 nd , 3 rd turns up	Flag: slope goes positive
1 st , 2 nd , 3 rd turns down	Flag: slope goes negative
1 st , 2 nd , 3 rd changes direction	Flag: line turns up or turns down
1 st , 2 nd , 3 rd near #3 +/- #4	True when study >= #3 - #4 and #2 <= #3 + #4
1 st , 2 nd , 3 rd between #3 & #4	True when study = #3 and study <= #4
1 st , 2 nd , 3 rd between #3 & (#3 + #4)	True when study >= #3 and study <= #3 + #4
1 st , 2 nd , 3 rd as percent of scale	100 * (Study value – Scale low) / Scale Range
1 st , 2 nd , 3 rd , 4 th , 5 th parameters	Parameters 1, 2, 3, Offset U/D, Shift L/R
Stop Studies	Description
Stop, High stop, Low stop value	Study value
Stop spread	Close – Study Value
Stop slope	Study value – prior study value
Stop as percent of scale	100 * (Study value – Scale low) / Scale Range
Stop hit, High stop hit, Low stop hit	Flag: true when stop is touched
Position long	Flag: Low stop is active
Position short	Flag: High stop is active
1 st , 2 nd , 3 rd parameters	Parameters 1, 2, Offset U/D
DYO Study	Description
Row A – L value	Study value
Row A – L <> 0	Flag: study value not equal to zero, Boolean True
Row A – L = 0	Flag: study value equal to zero, ie. Boolean False
Row A – L rising	Flag: study value >= prior study value
Row A – L falling	Flag: study value <= prior study value

Row A – L turns up	Flag: slope goes positive
Row A – L turns down	Flag: slope goes negative
Row A – L goes true	Flag: study flag goes from False to True
Row A – L goes false	Flag: study flag goes from True to False
Row A – L changes state	Flag: study flag changes state
Row A – L bars since true	Index – prior index where study flag was True
Row A – L bars since false	Index – prior index where study flag was False

Quote Value Category

Selection	Description
Last, High, Low, Open, Volume, Yesterday Close	Daily values from quote table
Open Interest	Applies only to futures symbols
Tick Count	Number of trade ticks in session
Tick Volume	Volume of last trade tick
Ask, Ask Size, Bid, Bid Size	Values from quote table
Net, abs(Net)	Net = Close – Yesterday Close, Absolute(Net)
Net Bid	Bid – Yesterday Close
Net High	Close - High
Net Low	Close - Low
Net Open	Close - Open
Range	Daily High - Low
True Range	True Range High – True Range Low
True Range High	Highest(High, Yesterday Close)
True Range Low	Lowest (Low, Yesterday Close)
Formula involving C, H, L, O	C = Close, H = High, L = Low, O = Open
% Range	100 * Net Low / Range
% Off High	100 * Net High / High
% Off Low	100 * Net Low / Low
% Off Open	100 * Net Open / Open
TickTime as hhhmss	Return tick time as a number in this format.
TickTime in second	Return tick time in seconds from Jan 1 st , 1970.
TickTime as TDateTime	Return tick time as a fraction of a 24 hour period.

Chart Value Category

Selection	Description
Index after Time #2	Bar Index at or after Time #2. #2 is entered as hhmm, no colon.
Bar Spacing in pixels	Number of pixels between bar centers
#2 Percent of Bar's Range	Price that is a percent of a bar's Range offset from bar's Low
#2 Percent of Scale	Price that is a percent of a Scale offset from Scale Low
#2 Minute Close, High, Low, Open	Encapsulate a time period in minutes
Prior #2 Minute Close, High, Low, Open	Encapsulate the prior time period in minutes
#2 Bar Close, High, Low, Open	#2 may be a bar count, or selection of Minute, Hour, Day, Week, Month, Quarter, Year. Encapsulates a set of bars.
Prior #2 Bar Close, High, Low, Open	Same as the above but returns the prior encapsulated period.
#2 Minute High, Low after 1 st , 2 nd Open	Highest High or Lowest Low in a time period after a session open
Earliest(H,L) in #2 min 1 st , 2 nd Open	Similar to the above but returns the earliest High or Low
Latest(H,L) in #2 min 1 st , 2 nd Open	Similar to the above but returns the latest High or Low
Minute, Hour, Day, Week, Month, Year(Now)	Return the selected information for the current date and time
Second of the Day(Now)	Convert the current time into seconds relative to midnight
Minute of the Day(Now)	Convert the current time into minutes relative to midnight
Minute of the Year(Now)	Convert the current time into minutes relative to Jan 1 st , this year
Day of Week(Now)	Sunday = 1 through Saturday = 7
Now as TDateTime	Integer part is Days from 1900. Fraction is Time. 0.5 is noon.
Time Session 1,2 Open, Close	Time of session starts and stops
Minutes in Session 1,2	Duration of the session in minutes
1 st Sun, Mon, Tue, Wed, Thu, Fri, Sat of Month	Day of the month for the 1 st selected weekday

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Chart Value	0	= #2 Bar Low		Week	0
		{			0
A	Chart Value	[0] := Week Bar Low	<input type="checkbox"/>		
B	Chart Value	[0] := Week Bar High	<input type="checkbox"/>		
C	Chart Value	[0] := Quarter Bar High	<input checked="" type="checkbox"/>		Study Value
D	Chart Value	[0] := Quarter Bar Low	<input checked="" type="checkbox"/>		Study Value

Branching Category

Selection	Description
if # then go ahead [#] rows	Conditionally skip over following rows
if # then go back [#] rows	Conditionally jump back to a prior row
if # then do next row	Conditionally execute the following row
if # then abort this DY0	Conditionally skip over all remaining rows
inc(V); if # then go back [#] rows	Increase Variable, then evaluate expression
dec(V); if # then go back [#] rows	Decrease Variable, then evaluate expression
for V := #3 to #4 do next [#] rows	FOR loop, increment V from #3 to #4 and do rows
while # do next [#] rows	WHILE loop, do [#] rows as long as # is True
repeat next [#] rows until #	REPEAT loop, do [#] rows as long as # is False
if # then abort all calculations	Conditionally skip calculation of all studies and DYOs. Probably use with Resize Event Flag to prevent a recalculation

ESPL and DLL Categories

Selection	Description
V := Label(Index, V, #2, #3, #4)	Call function for every bar. Result assigned to V. Label is the function name. If label is blank name is double row letter, ex. AA Pass parameters: Index, Variable, fields #2, #3, #4.
if Index >= Bar Count - [#] then	Conditionally call function
if Index in Visible Bars then	Conditionally call function for visible bars only
if Index = 1st Visible Bar then	Call function one time at 1 st visible bar
if Index = [#] then	Call function when Index = [#]
if Index = #2 then	Call function when Index = #2
if #2 then	Conditionally call function when #2 is True
if # then	Conditionally call function when # is True
if ## then	Conditionally call function when ## is True

Reference Tags

	Description
[A] through [L]	Return a DYO row value
[@A] through [@L]	Return a DYO row label, for use with Action Cell := messag
[@M]	Return a date string in the format of mmm d. Example: Feb 5
[0] through [999], or [variable name]	Return a Variable value. See Formatting in the next table.
Chart and Bar Tags	Description
[\$S], [\$U]	Chart symbol, Chart file name
[\$C], [\$H], [\$L], [\$O]	Bar Close, High, Low, Open
[\$V], [\$T]	Bar Volume, Tick Count
[\$N], [\$R]	Bar Net, Range
[\$A], [\$B]	Ask Volume, Bid Volume
[\$E], [\$F]	Ask Ratio, Bid Ratio
[\$J], [\$K]	Buy Pressure, Sell Pressure
[\$W], [\$X]	Buy Ratio, Sell Ratio
[\$%]	$(C - L) / (H - L)$
[\$2]	$(H + L) / 2$
[\$3]	$(H + L + C) / 3$
[\$4]	$(O + H + L + C) / 4$
[\$I]	Bar Index
[\$M]	Chart time frame
[\$Z], [\$Y]	Tick Size, Tick Size / 2
[\$Q]	Tick field from Symbol Properties.
[\$G]	Min. Grid from Symbol Properties.
[\$P]	Leverage to convert points to \$
[\$D]	Bar time in a format of hhmm or date in a format of yyymmdd
[\$<], [\$>]	Day session Open, Close time hhmm
[\$I], [\$]	2nd session Open, Close time hhmm
[\$@]	Computer clock in hh:mm:ss format
[\$#]	Layer number
[\$*]	Alert message generated by DYO Action statement
Quote Page Tags	Description
[^C], [^H], [^L], [^O]	Daily Close, High, Low, Open
[^V], [^I]	Daily Volume, Open Interest
[^N], [^Y]	Daily Net, Yesterday's Close
[^A], [^D], [^E]	Ask, Ask Size, Tick Vol at Ask

[^B], [^F], [^G]	Bid, Bid Size, Tick Vol at Bid
[^%]	$(C - L) / (H - L)$
[^2]	$(H + L) / 2$
[^3]	$(H + L + C) / 3$
[^4]	$(O + H + L + C) / 4$
[^R]	Range = High - Low
[^S]	True Range = Highest(H, YC) – Lowest(L, YC)
[^T], [^Z]	Tick Volume, Between Bid Ask
[^W]	Word description. Example: INTERNATIONAL BUSINESS MAC

Formatting

Character	Description	Example
\$	Add a \$ suffix to format the value as a price using the chart's scale.	[A\$]
.	Add a period suffix to format the value as an integer.	[\$V.]
.#	Add a period suffix and the number of decimals to show.	[A.2]
,	Add a comma suffix to format the value as an integer with commas for large values.	[\$V,]
,#	Add a comma suffix and the number of decimals to show. Use with large values.	[A,2]
`	The ` will add a degree symbol. ` shares the ~ key and follows the [].	[A`]
	Format as Boolean to show the word 'True' or the word 'False'.	[1]
@	Add a @ suffix to format the value as an ASCII character. Example: Variable [1] has a value of 66. [1@] would return the letter 'B'	[1@]
text1 text2	Format as Boolean and change the words to be shown for True and False.	[1 Up Down]

Markers

177 Marker selections are available for use in DYO's, Studies and Draw Tools. They can create special visual effects.

None - Select the 1st entry of None when you do not want to display a Marker.

Curves - The 1st eight selections would be used for plotting a study curve. The Rise/Fall selections use 2 colors which come from the row with the marker selection and the row which follows. The 1st color is used when the study curve is ascending, and the 2nd color is used when the study curve is descending.


The line thickness can be 1, 2 or 3 pixels in width.

Example

Arrows - The chevrons, large arrows and small arrows will draw relative to the marker's point.

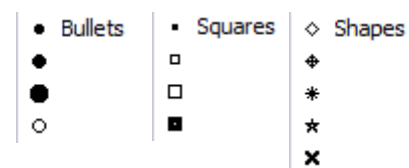
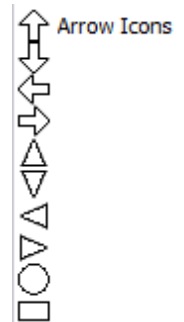
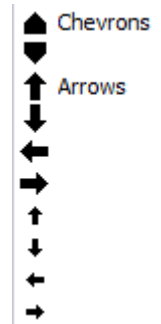
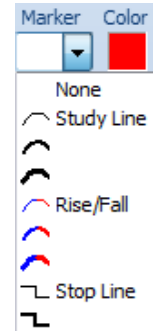
Arrow Icons – The arrow icons differ from the arrow markers in that a frame is drawn in black and the interior is filled with the selected color. The default size is 16 pixels.

For the 'Action | if ## then marker in column #2' selection only, the Offset entry controls the size of these 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.

Size examples: 

Bands - 4 block markers of different thicknesses are available. These blocks will draw full bar width so that they make a continuous band when used on a fixed Marker Location such as Top Row 1. [Example](#)

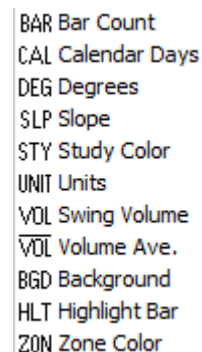
Shapes - A large variety of shapes are available. They include bullets, pegs, squares, stars, digits, upper and lower case letters. These markers could be used in lieu of a line in plotting a study. [Example](#)



Bar Styles - Four markers are available that can draw a Bar, Candlestick, Ensign Rocket, or Ensign Flute. The marker needs 4 values in the order of High, Low, Open, and Close on 4 adjacent DYO rows. Place the marker on the 4th row with the Close value. [Encapsulation](#) [Example 2](#) [Example 3](#)



BAR - Show the number of bars between the 2 ends of a draw line. [Example](#)



CAL - Show the number of calendar days between the 2 ends of a draw line. For use on a daily chart.

DEG - Show the slope of the line in Degrees, like reading an angle on a protractor held to the screen.

SLP - Shows the Slope of a line segment.

STY - A Study Alert can use this marker to change the line color of a referenced Study.

UNIT - Show the length of a diagonal draw line in Units of the bar spacing.

VOL & Ave VOL - These 2 markers sum the bar volumes in the span between the 2 ends of a Draw Line. The Average VOL marker will divide the sum by the number of bars in the span.

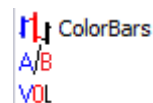
BGD - Sets the background color of the chart or of the sub-window. Use cautiously as this can be CPU intensive. [Example](#)

HLT - Show a rectangle highlight behind the bar. [Example](#)

ZON - Show a vertical zone stripe from the bottom of the chart to the top of the chart. This can be used to color sections of the chart background.

[Example 1](#) [Example 2](#)

Color Bars - Sets the color for the chart's bar or candlestick. Have the Color Bar study selection on Normal so the coloring done by this marker does not conflict or compete with a Color Bar study.



A/B - Sets the color for the Ask/Bid sub-window histogram bar.

VOL - Sets the color for the Volume histogram bar in the volume sub-window.

[Example 1](#) [Example 2](#) [Example 3](#)

LBL - Use the DY0's Label text as the Marker. This can create interesting effects with a label like "<---->". In the Pesevento Patterns study, LBL will label the swing points with HH, LH, LL, and HL. [Example 1](#) [Example 2](#)

D:T - Show a bar's date on a daily, weekly, or monthly chart. Show a bar's Time on an intra-day chart.

Delta \$ - Show the change in price between the 2 ends of a draw line. [Example](#)

% \$ - Show both a percentage and a price.








% - Show a percentage. Example use would be as the right side marker for the Fibonacci Levels draw tool.

\$ - Show a price. Example use would be as the right side marker for a Horizontal draw line. [Example](#)

Dollar Delta – Show the price change in dollars between the 2 ends of a draw line. The price change is multiplied by the leverage. [Example](#)

LBL	Label is Token
D:T	Date or Time
Δ\$	Price Change
⌘\$	Percent & Price
⌘	Percent
\$	Price
\$Δ	Dollar Change

Image List

												
0	1	2	3	4	5	6	7	8	9	10	11	12
												
13	14	15	16	17	18	19	20	21	22	23	24	25
												
26	27	28	29	30	31	32	33	34	35	36	37	38
												
39	40	41	42	43	44	45	46	47	48	49	50	51
												
52	53	54	55	56	57	58	59	60	61	62	63	64
												
65	66	67	68	69	70	71	72	73	74	75	76	77
												
78	79	80	81	82	83	84	85	86	87	88	89	90
												
91	92	93	94	95	96	97	98	99	100	101	102	103
												
104	105	106	107	108	109	110	111	112	113	114	115	116
												
117	118	119	120	121	122	123	124	125	126	127	128	129
												
130	131	132	133	134	135	136	137	138	139	140	141	142
												
143	144	145	146	147	148	149	150	151	152	153	154	155
												
156	157	158	159	160	161	162	163	164	165	166	167	168
												
169	170	171	172	173	174	175	176	177	178	179	180	181
												
182	183	184	185	186	187							
												

Example: if [301] Button1.Image := 171 else 172; {sets the image of a toolbar button1 to the green or red check mark}