Appendix

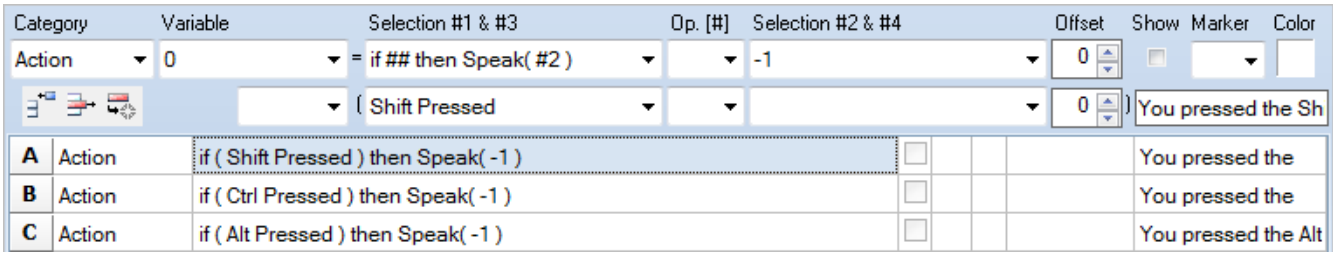
Expression Data Values

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 1st session values (Day Session)
Session 2 Close, High, Low	A bar's 2nd 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, Day of the Day or Year	Values for a bar's time stamp from midnight or from Jan 1st.
Minute, Hour, Day, Week, Month, Quarter, Year	Values for a bar's time stamp
Day of Week, Day of Week (Now)	Bar's or Now's time stamp: 1 = Sunday through 7 = Saturday
Second, Minute, Day of the Day, Week, Year (Now)	Values for Now from midnight or from Jan 1st.
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
Chart Has Focus	True when chart has focus or is the surface in a stack with focus.
Sub-window Scale High, Low, Midpoint, Range	Returns scale values for the DYO's sub-window 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 >= Last Visible Bar	True when the Index is at or beyond the index of the Last Visible Bar
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 1st 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 1st session (Day) or 2nd 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
Alphabetical Index	All charts on the visible layer are indexed alphabetically. Useful as the Row parameter for Spreadsheet statements.
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



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

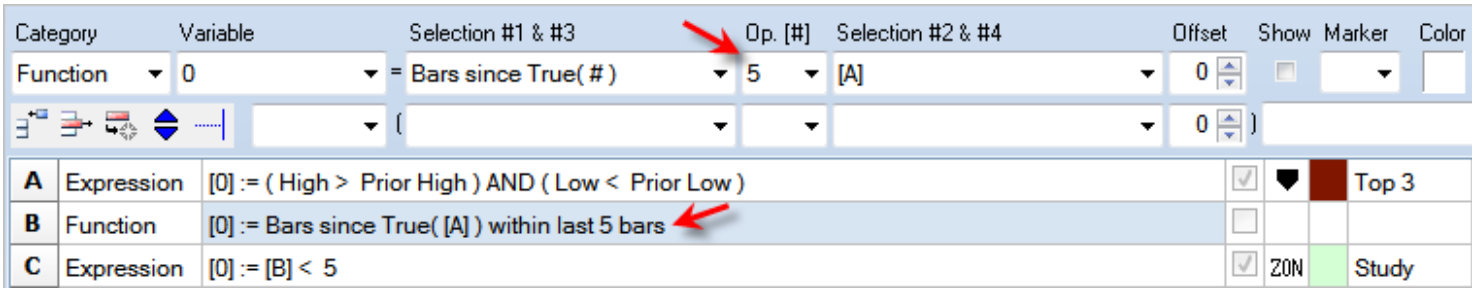
Math Operators	Description
+, -, *, /, div	Add, subtract, multiply, divide, integer division
^	Power, ex: 5 ^ 2 = 25, 5 ^ 3 = 125
* -, / -	Multiply or divide a negative value
mod	Modulo is the remainder from a division returned as an integer
Boolean Operators	**Description**
AND, AND Not, OR, XOR, Not	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

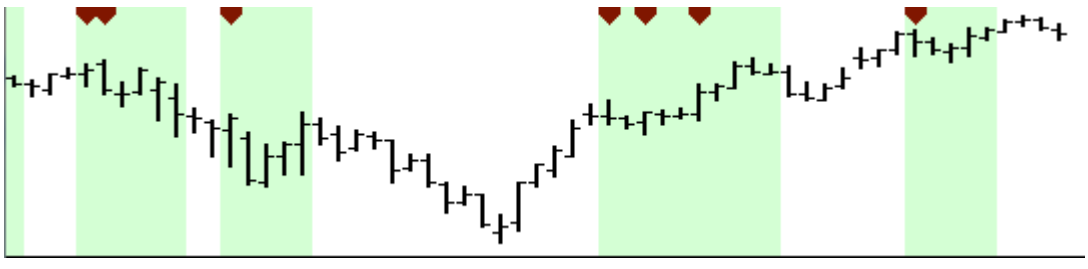
Function Category

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 [#]
Sum DYOs(#2, modulo #3, offset #4)	Sums the #2 row in a named DYO from multiple charts.
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(sqr(#2) + sqr(##))	Pythagorean theorem. Find hypotenuse from 2 sides.
Range Percent(#2, [#], #3, #4)	Returns #3 / 100 * range of the #2 set of size [#], or #4 minimum.
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 Add a value in the Op.[#] field to limit the search's distance back
Bars since False(#)	Current Index – prior Index where # is False Add a value in the Op.[#] field to limit the search's distance back
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 Add a value in the Op.[#] field to limit the search's distance back
Index of last False(#)	Recent Index where # was False Add a value in the Op.[#] field to limit the search's distance back
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 or covers prior #2)	
Index(Range inside or 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

Example Bars Since True: Row C test is for recent 4 bars, so limit the Bars Since True to last 5 bars.





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 1st bar with the operator relationship to time #2
1st tick of New bar	True for 1st tick that starts a new bar
1st bar of New Minute	True for 1st bar of a new minute period
1st bar of New Hour + #2 min	True for 1st bar of a new hour period, offset #2
1st bar of New Day, or New Trade Day	True for 1st bar of a new calendar date, or new trade day
1st bar of Session 1, 2	True for 1st bar of 1st session, or of 2nd session
1st bar of Either Session	True for 1st bar of a new session
1st bar of New Week, Month, Quarter, Year	True for 1st 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 1st or 2nd 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 1st session close, or 2nd 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, Start = #3, Stop = #4	Change the Section Font Size. Bold = negative Optional location.
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
If # then Do(Message)	The Message actions are the same as those available in the Scheduler
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 these examples: Trading System Trend Runner System
if ## then Sell Stop Price := #2	
if # then Exit Position, Exit Long, Exit Short Pos.	
if # then Cancel Buy Limit, Cancel Sell Limit	
if # then Cancel Buy Stop, 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.
If # then Fill Trade Array(#2)	Fill a variable array starting at index #2. with 28 trade values. See below.
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) if # then OutputTop(Message)	Write Label or Message to the bottom of the Output Window. OutputTop will insert the Text at the top of the Output Window.
if ## then Output(#2) if ## then OutputTop(#2)	Text is the Variable #2 name which can be lengthy. Ouput adds to the bottom. OutputTop inserts at the top.

if # then Send(Message)	Send Label or Message to another application.
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 Output Window.Open .Close .Clear	Conditionally control the Output Window
if # then Output Log.Load .Save .Delete	Conditionally control the OutputLog.txt file in C:\Ensign10 folder.
if ## then Output Filename = Variable #2 Name	Conditionally set the Output Filename using Variable 2's name
if # then E-mail Message if ## then E-mail(#2)	Conditionally e-mail the Message text E-mail text is the Variable #2 name which can be lengthy.
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 [#] Example: Shift Right([20], 5) 1st variable will be zeroed.
if ## then Shift Left(#2, [#])	Shift leftward a set of Variables. Last variable will be zeroed.
if ## then Rotate Right(#2, [#])	Rotate rightward a set of Variables. Example: Rotate([20], 5) Last value will move to the 1st position of the variable set.
if ## then Rotate Left(#2, [#])	Rotate leftward a set of Variables. 1st 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 oper 0 then Button[#].Image := #3 else #4	Change the button image on the 6 toolbar buttons. [#] is 1..6

If #2 oper 0 then Marker(#3, #4)	Conditionally create a Marker object at #3 Index and #4 Price. The marker is the selected token, color is the color panel, and text is the DYO row's message. The text will show on the right hand side of the marker.
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

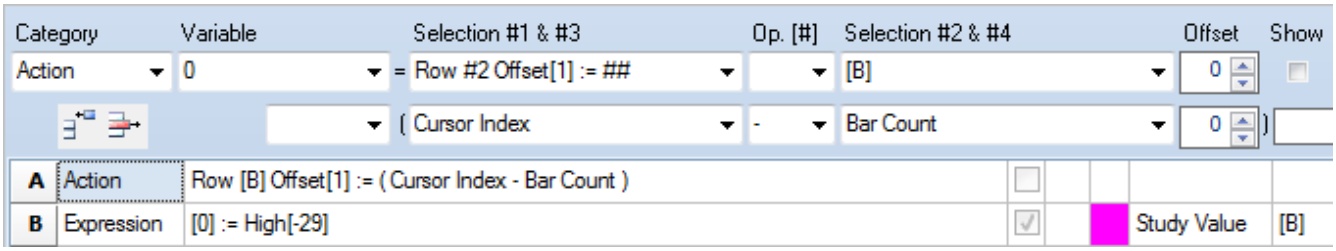
PriceFinder selections evaluate a Flag and return the price where the Flag will be True, False, or Change states. For example, PriceFinder can determine the price that would make two moving averages cross, or cause CCI to cross zero, or make a bar reach a channel band.

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 use for this statement is to preserve values that are received in real-time from a quote page.

Row #2 Offset[1] :=

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



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

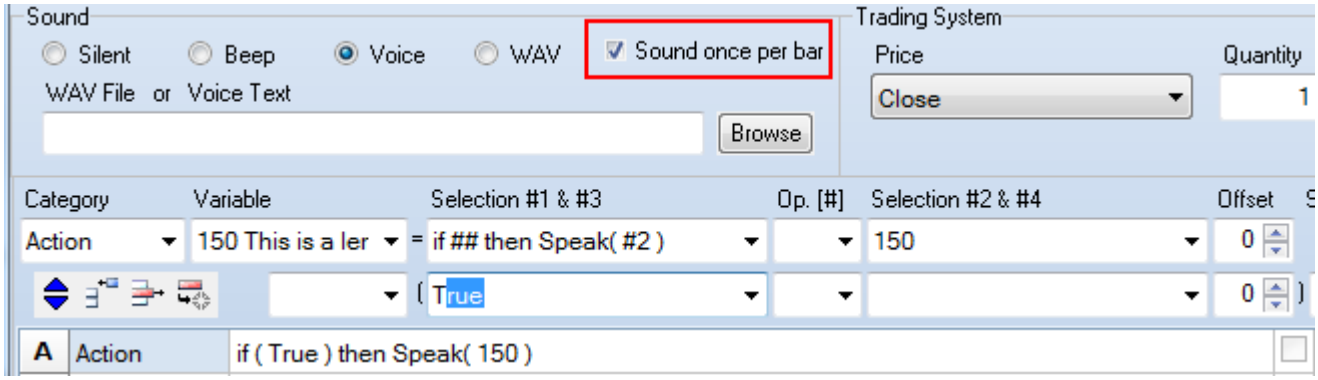
if ## then Speak(#2)

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.



Select the Voice bullet. 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 OutputTop(Message)
if ## then Output(#2)
if ## then OutputTop(#2)

The text string is written in the Output Window. These statements can be very useful to document the execution flow through the DYOs for debugging purposes. It is recommended that the statement only be executed when Bar Index = Bar Count to avoid a repetitious writing of information to the Output Window when a chart resizes and recalculates across all bars.

Output appends text to the bottom of the Output Window. OutputTop inserts text at the top of the Output Window.

When the line count reaches 1,500 lines, the most recent 1,000 lines are retained.

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.

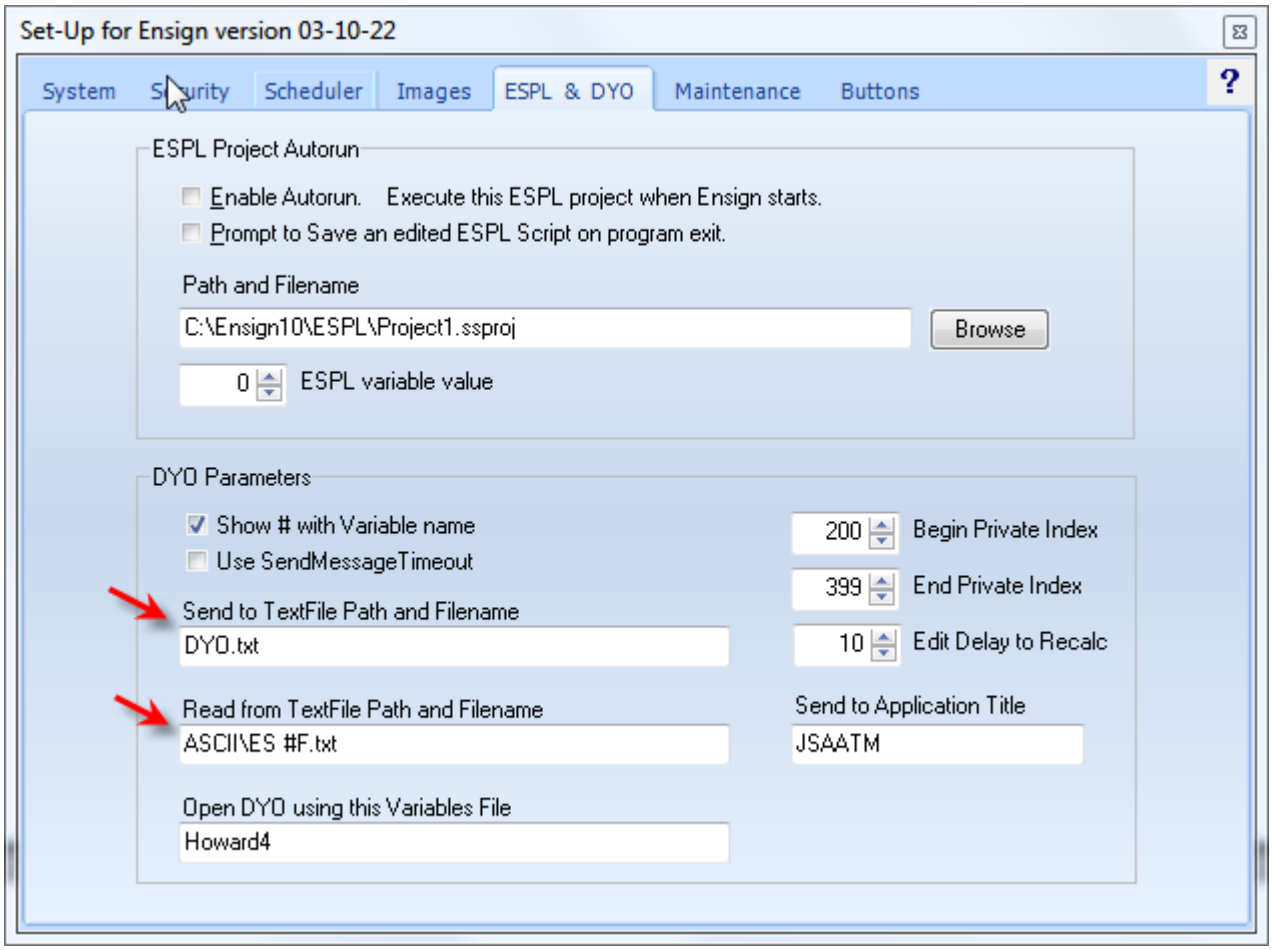
The text may contain reference tags such as [$S] and [$C], which will be resolved before the text appended.

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.



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 E-mail(#2)
If # then Text Message

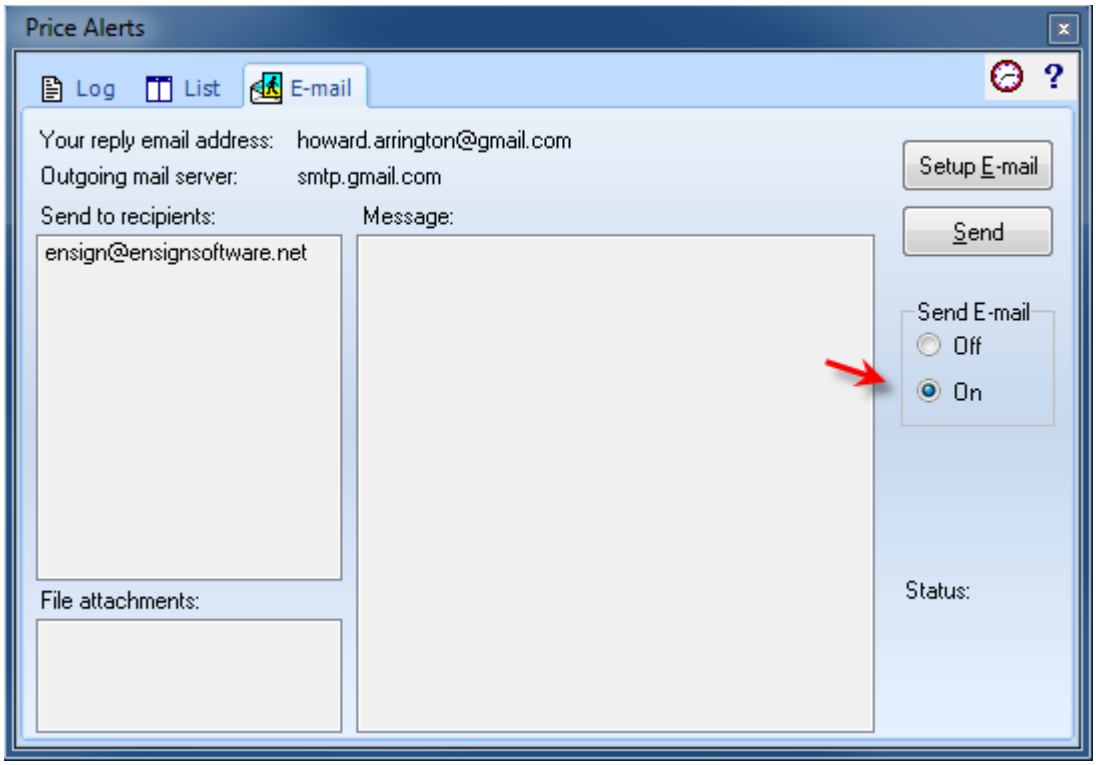
Message is a text string from the label on the DYO row. If the label is blank, then Message is the text from the Message Text edit box. The E-mail statement will automatically insert the chart's symbol as a prefix to the Message text.

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.

These 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 DYOs.



If # then Fill Trade Array(#2)
If # then Reset Day Stats

The #2 value is a Variable index. 29 values from the trading system will fill 29 sequential variables starting with variable [#2]. For example, when the #2 field has a 100 entry, variables 100 through 128 would contain the following information from the trading system.

Measurement of the Max Day Drawdown and Max Day Position are from the reset of the day stats.

Index	Value	Index	Value	Index	Value
100	Position	110	Last Profit	120	Last Trade Position
101	Prior Position	111	Max Day Drawdown	121	Bar Index
102	Quantity	112	Win Trades	122	Trade Index
103	Entry Price	113	Loss Trades	123	Open Price
104	Exit Price	114	Max Chart Position	124	High Price
105	Win Profit	115	Max Day Position	125	Low Price
106	Loss Profit	116	Buy Limit	126	Close Price
107	Closed Profit	117	Buy Stop	127	Bar Date
108	Open Profit	118	Sell Limit	128	Average Entry Price
109	Total Profit	119	Sell Stop		

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

Chart Actions

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

Keyboard Actions

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

Study Category

Most Studies	Description
1st, 2nd, 3rd line value	Study value
1st, 2nd, 3rd line slope	Study value – prior study value
1st line >= 2nd line	Flag: 1st line greater than or equal to 2nd line
1st line <= 2nd line	Flag: 1st line less than or equal to 2nd line
1st line X> 2nd line	Flag: 1st line cross above 2nd line
1st line X< 2nd line	Flag: 1st line cross below 2nd line
1st line X<> 2nd line	Flag: 1st line crosses 2nd line
1st line and 2nd line rising	Flag: both line slopes are positive
1st line and 2nd line falling	Flag: both line slopes are negative
1st, 2nd, 3rd line rising	Flag: study value => prior study value
1st, 2nd, 3rd line falling	Flag: study value <= prior study value
1st, 2nd, 3rd turns up	Flag: slope goes positive
1st, 2nd, 3rd turns down	Flag: slope goes negative
1st, 2nd, 3rd changes direction	Flag: line turns up or turns down
1st, 2nd, 3rd near #3 +/- #4	True when study >= #3 - #4 and #2 <= #3 + #4
1st, 2nd, 3rd between #3 & #4	True when study = #3 and study <= #4
1st, 2nd, 3rd between #3 & (#3 + #4)	True when study >= #3 and study <= #3 + #4
1st, 2nd, 3rd as percent of scale	100 * (Study value – Scale low) / Scale Range
1st, 2nd, 3rd, 4th, 5th 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
1st, 2nd, 3rd 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

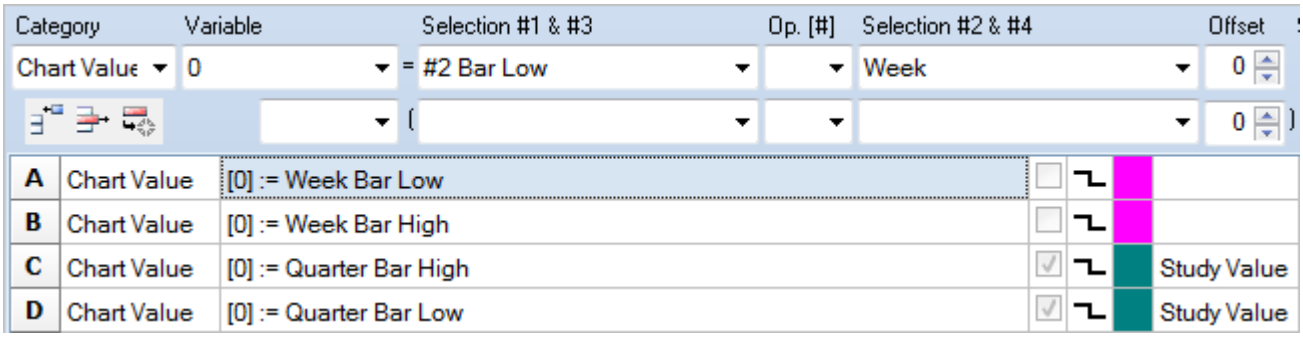
Row A – L turns down	Flag: slope goes negative
Row A – L goes true	Flag: study flag goes from False to True
Row A – L goes false	Flag: study flag goes from True to False
Row A – L changes state	Flag: study flag changes state
Row A – L bars since true	Index – prior index where study flag was True
Row 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 hhmmss	Return tick time as a number in this format.
TickTime in second	Return tick time in seconds from Jan 1st, 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 1st, 2nd Open	Highest High or Lowest Low in a time period after a session open
Earliest(H,L) in #2 min 1st,2nd Open	Similar to the above but returns the earliest High or Low
Latest(H,L) in #2 min 1st,2nd 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 1st, 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
1st Sun, Mon, Tue, Wed, Thu, Fri, Sat of Month	Day of the month for the 1st selected weekday
ColorBar Bar, Ask/Bid, Volume(#2)	Return the colorbar state at the #2 Index. 1 = Bullish, 2 = Bearish
Color of Bar, Ask/Bid, Volume(#2)	Return the color value at the #2 Index. 0 = Black, 255 = Red.



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 DYO	Conditionally skip over all remaining rows
inc(V); if # then go back [#] rows	Increment Variable, then evaluate expression
dec(V); if # then go back [#] rows	Decrement 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 1st 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], [$s]	Chart symbol, Chart file name, Chart symbol with month resolved for #F
[$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 $
[$D]	Bar time in a format of hhmm or date in a format of yymmdd
[$D:]	Bar time is returned as a string in the format of hh:mm
[$\]	Bar date in a format of mmdd. This is a numeric value.
[$\-]	Bar date is returned as a string in the format of mm-dd
[$\/]	Bar date is returned as a string in the format of mm-dd-yy
[$\<] or [$D<]	Bar date Day of Week is returned as a letter: S, M, T, W, T, F, S
[$\=] or [$D=]	Bar date Day of Week returns an abbreviation: Sun, Mon, Tue, Wed, Thu
[$\>] or [$D>]	Bar date Day of Week returns its name: Sunday, Monday, Tuesday, etc.
[$<], [$>]	Day session Open, Close time hhmm
[$(], [$)]	2nd session Open, Close time hhmm
[$@]	Computer clock is returned as a string in the format of hh:mm:ss
[$0]	Computer clock in yyyymmddhhnn0 format. Note ending zero.

[$#]	Layer number
[$*]	Alert message generated by DYO Action statement
[%]	Level value for an alert on Fib Levels, Retracement, and Ratio Level
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.	[$V**,**]
,#	Add a comma suffix and the number of decimals to show. Use with large values.	[A,2]
`	The ` will add a degree symbol. ` shares the ~ key and follows the [ ].	[A]`
*	Format as Boolean to show the word 'True' or the word 'False'.	[1*]
@	Add a @ suffix to format the value as an ASCII character. Example: Variable [1] has a value of 66. [1@] would return the letter 'B'	[1@]
:	Add a : suffix to format the value as a Time with format of hh-mm Add a : prefix to convert a Minutes of the Day value into a Time with format of hh:mm	[$D:] [:5]
-	Add a - suffix to format the value as a Date with format of mm-dd	[$\-]
/	Add a / suffix to format the value as a Date with format of mm-dd-yy	[1/]
|text1|text2	Format as Boolean and change the words to be shown for True and False.	[1|Up|Down]
!	Add a ! suffix to return a string from the variable's name.	[1!]
#	Add a # suffix to use the variable's value as the index of the variable's name to return.	[1#]
<	Add < after the] bracket to align left the text in a spreadsheet cell. Adds HTML formatting.	[210.$]<
^	Add ^ after the] bracket to center the text in a spreadsheet cell. Adds HTML formatting.	[210.$]^
>	Add > after the] bracket to align right the text in a spreadsheet cell. Adds HTML formatting.	[210.$]>
<#	Add < and # ahead of the] bracket to align left the text in a width of # characters. Spaces pad.	[201.<5]
^#	Add ^ and # ahead the] bracket to center the text in a width of # characters. Spaces pad.	[201.^5]
>#	Add > and # ahead the] bracket to align right the text in a width of # characters. Spaces pad.	[201.>5]