// This work is licensed under a Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) https://creativecommons.org/licenses/by-nc-sa/4.0/

// © LuxAlgo

//@version=5

indicator("Smart Money Concepts [LuxAlgo]", "Smart Money Concepts [LuxAlgo]"

, overlay = true

, max\_labels\_count = 500

, max\_lines\_count = 500

, max\_boxes\_count = 500

, max\_bars\_back = 500)

//-----------------------------------------------------------------------------{

//Constants

//-----------------------------------------------------------------------------{

color TRANSP\_CSS = #ffffff00

//Tooltips

string MODE\_TOOLTIP = 'Allows to display historical Structure or only the recent ones'

string STYLE\_TOOLTIP = 'Indicator color theme'

string COLOR\_CANDLES\_TOOLTIP = 'Display additional candles with a color reflecting the current trend detected by structure'

string SHOW\_INTERNAL = 'Display internal market structure'

string CONFLUENCE\_FILTER = 'Filter non significant internal structure breakouts'

string SHOW\_SWING = 'Display swing market Structure'

string SHOW\_SWING\_POINTS = 'Display swing point as labels on the chart'

string SHOW\_SWHL\_POINTS = 'Highlight most recent strong and weak high/low points on the chart'

string INTERNAL\_OB = 'Display internal order blocks on the chart\n\nNumber of internal order blocks to display on the chart'

string SWING\_OB = 'Display swing order blocks on the chart\n\nNumber of internal swing blocks to display on the chart'

string FILTER\_OB = 'Method used to filter out volatile order blocks \n\nIt is recommended to use the cumulative mean range method when a low amount of data is available'

string SHOW\_EQHL = 'Display equal highs and equal lows on the chart'

string EQHL\_BARS = 'Number of bars used to confirm equal highs and equal lows'

string EQHL\_THRESHOLD = 'Sensitivity threshold in a range (0, 1) used for the detection of equal highs & lows\n\nLower values will return fewer but more pertinent results'

string SHOW\_FVG = 'Display fair values gaps on the chart'

string AUTO\_FVG = 'Filter out non significant fair value gaps'

string FVG\_TF = 'Fair value gaps timeframe'

string EXTEND\_FVG = 'Determine how many bars to extend the Fair Value Gap boxes on chart'

string PED\_ZONES = 'Display premium, discount, and equilibrium zones on chart'

//-----------------------------------------------------------------------------{

//Settings

//-----------------------------------------------------------------------------{

//General

//----------------------------------------{

mode = input.string('Historical'

, options = ['Historical', 'Present']

, group = 'Smart Money Concepts'

, tooltip = MODE\_TOOLTIP)

style = input.string('Colored'

, options = ['Colored', 'Monochrome']

, group = 'Smart Money Concepts'

, tooltip = STYLE\_TOOLTIP)

show\_trend = input(false, 'Color Candles'

, group = 'Smart Money Concepts'

, tooltip = COLOR\_CANDLES\_TOOLTIP)

//----------------------------------------}

//Internal Structure

//----------------------------------------{

show\_internals = input(true, 'Show Internal Structure'

, group = 'Real Time Internal Structure'

, tooltip = SHOW\_INTERNAL)

show\_ibull = input.string('All', 'Bullish Structure'

, options = ['All', 'BOS', 'CHoCH']

, inline = 'ibull'

, group = 'Real Time Internal Structure')

swing\_ibull\_css = input(#089981, ''

, inline = 'ibull'

, group = 'Real Time Internal Structure')

//Bear Structure

show\_ibear = input.string('All', 'Bearish Structure'

, options = ['All', 'BOS', 'CHoCH']

, inline = 'ibear'

, group = 'Real Time Internal Structure')

swing\_ibear\_css = input(#f23645, ''

, inline = 'ibear'

, group = 'Real Time Internal Structure')

ifilter\_confluence = input(false, 'Confluence Filter'

, group = 'Real Time Internal Structure'

, tooltip = CONFLUENCE\_FILTER)

internal\_structure\_size = input.string('Tiny', 'Internal Label Size'

, options = ['Tiny', 'Small', 'Normal']

, group = 'Real Time Internal Structure')

//----------------------------------------}

//Swing Structure

//----------------------------------------{

show\_Structure = input(true, 'Show Swing Structure'

, group = 'Real Time Swing Structure'

, tooltip = SHOW\_SWING)

//Bull Structure

show\_bull = input.string('All', 'Bullish Structure'

, options = ['All', 'BOS', 'CHoCH']

, inline = 'bull'

, group = 'Real Time Swing Structure')

swing\_bull\_css = input(#089981, ''

, inline = 'bull'

, group = 'Real Time Swing Structure')

//Bear Structure

show\_bear = input.string('All', 'Bearish Structure'

, options = ['All', 'BOS', 'CHoCH']

, inline = 'bear'

, group = 'Real Time Swing Structure')

swing\_bear\_css = input(#f23645, ''

, inline = 'bear'

, group = 'Real Time Swing Structure')

swing\_structure\_size = input.string('Small', 'Swing Label Size'

, options = ['Tiny', 'Small', 'Normal']

, group = 'Real Time Swing Structure')

//Swings

show\_swings = input(false, 'Show Swings Points'

, inline = 'swings'

, group = 'Real Time Swing Structure'

, tooltip = SHOW\_SWING\_POINTS)

length = input.int(50, ''

, minval = 10

, inline = 'swings'

, group = 'Real Time Swing Structure')

show\_hl\_swings = input(true, 'Show Strong/Weak High/Low'

, group = 'Real Time Swing Structure'

, tooltip = SHOW\_SWHL\_POINTS)

//----------------------------------------}

//Order Blocks

//----------------------------------------{

show\_iob = input(true, 'Internal Order Blocks'

, inline = 'iob'

, group = 'Order Blocks'

, tooltip = INTERNAL\_OB)

iob\_showlast = input.int(5, ''

, minval = 1

, inline = 'iob'

, group = 'Order Blocks')

show\_ob = input(false, 'Swing Order Blocks'

, inline = 'ob'

, group = 'Order Blocks'

, tooltip = SWING\_OB)

ob\_showlast = input.int(5, ''

, minval = 1

, inline = 'ob'

, group = 'Order Blocks')

ob\_filter = input.string('Atr', 'Order Block Filter'

, options = ['Atr', 'Cumulative Mean Range']

, group = 'Order Blocks'

, tooltip = FILTER\_OB)

ibull\_ob\_css = input.color(color.new(#3179f5, 80), 'Internal Bullish OB'

, group = 'Order Blocks')

ibear\_ob\_css = input.color(color.new(#f77c80, 80), 'Internal Bearish OB'

, group = 'Order Blocks')

bull\_ob\_css = input.color(color.new(#1848cc, 80), 'Bullish OB'

, group = 'Order Blocks')

bear\_ob\_css = input.color(color.new(#b22833, 80), 'Bearish OB'

, group = 'Order Blocks')

//----------------------------------------}

//EQH/EQL

//----------------------------------------{

show\_eq = input(true, 'Equal High/Low'

, group = 'EQH/EQL'

, tooltip = SHOW\_EQHL)

eq\_len = input.int(3, 'Bars Confirmation'

, minval = 1

, group = 'EQH/EQL'

, tooltip = EQHL\_BARS)

eq\_threshold = input.float(0.1, 'Threshold'

, minval = 0

, maxval = 0.5

, step = 0.1

, group = 'EQH/EQL'

, tooltip = EQHL\_THRESHOLD)

eq\_size = input.string('Tiny', 'Label Size'

, options = ['Tiny', 'Small', 'Normal']

, group = 'EQH/EQL')

//----------------------------------------}

//Fair Value Gaps

//----------------------------------------{

show\_fvg = input(false, 'Fair Value Gaps'

, group = 'Fair Value Gaps'

, tooltip = SHOW\_FVG)

fvg\_auto = input(true, "Auto Threshold"

, group = 'Fair Value Gaps'

, tooltip = AUTO\_FVG)

fvg\_tf = input.timeframe('', "Timeframe"

, group = 'Fair Value Gaps'

, tooltip = FVG\_TF)

bull\_fvg\_css = input.color(color.new(#00ff68, 70), 'Bullish FVG'

, group = 'Fair Value Gaps')

bear\_fvg\_css = input.color(color.new(#ff0008, 70), 'Bearish FVG'

, group = 'Fair Value Gaps')

fvg\_extend = input.int(1, "Extend FVG"

, minval = 0

, group = 'Fair Value Gaps'

, tooltip = EXTEND\_FVG)

//----------------------------------------}

//Previous day/week high/low

//----------------------------------------{

//Daily

show\_pdhl = input(false, 'Daily'

, inline = 'daily'

, group = 'Highs & Lows MTF')

pdhl\_style = input.string('⎯⎯⎯', ''

, options = ['⎯⎯⎯', '----', '····']

, inline = 'daily'

, group = 'Highs & Lows MTF')

pdhl\_css = input(#2157f3, ''

, inline = 'daily'

, group = 'Highs & Lows MTF')

//Weekly

show\_pwhl = input(false, 'Weekly'

, inline = 'weekly'

, group = 'Highs & Lows MTF')

pwhl\_style = input.string('⎯⎯⎯', ''

, options = ['⎯⎯⎯', '----', '····']

, inline = 'weekly'

, group = 'Highs & Lows MTF')

pwhl\_css = input(#2157f3, ''

, inline = 'weekly'

, group = 'Highs & Lows MTF')

//Monthly

show\_pmhl = input(false, 'Monthly'

, inline = 'monthly'

, group = 'Highs & Lows MTF')

pmhl\_style = input.string('⎯⎯⎯', ''

, options = ['⎯⎯⎯', '----', '····']

, inline = 'monthly'

, group = 'Highs & Lows MTF')

pmhl\_css = input(#2157f3, ''

, inline = 'monthly'

, group = 'Highs & Lows MTF')

//----------------------------------------}

//Premium/Discount zones

//----------------------------------------{

show\_sd = input(false, 'Premium/Discount Zones'

, group = 'Premium & Discount Zones'

, tooltip = PED\_ZONES)

premium\_css = input.color(#f23645, 'Premium Zone'

, group = 'Premium & Discount Zones')

eq\_css = input.color(#b2b5be, 'Equilibrium Zone'

, group = 'Premium & Discount Zones')

discount\_css = input.color(#089981, 'Discount Zone'

, group = 'Premium & Discount Zones')

//-----------------------------------------------------------------------------}

//Functions

//-----------------------------------------------------------------------------{

n = bar\_index

atr = ta.atr(200)

cmean\_range = ta.cum(high - low) / n

//HL Output function

hl() => [high, low]

//Get ohlc values function

get\_ohlc()=> [close[1], open[1], high, low, high[2], low[2]]

//Display Structure function

display\_Structure(x, y, txt, css, dashed, down, lbl\_size)=>

structure\_line = line.new(x, y, n, y

, color = css

, style = dashed ? line.style\_dashed : line.style\_solid)

structure\_lbl = label.new(int(math.avg(x, n)), y, txt

, color = TRANSP\_CSS

, textcolor = css

, style = down ? label.style\_label\_down : label.style\_label\_up

, size = lbl\_size)

if mode == 'Present'

line.delete(structure\_line[1])

label.delete(structure\_lbl[1])

//Swings detection/measurements

swings(len)=>

var os = 0

upper = ta.highest(len)

lower = ta.lowest(len)

os := high[len] > upper ? 0 : low[len] < lower ? 1 : os[1]

top = os == 0 and os[1] != 0 ? high[len] : 0

btm = os == 1 and os[1] != 1 ? low[len] : 0

[top, btm]

//Order block coordinates function

ob\_coord(use\_max, loc, target\_top, target\_btm, target\_left, target\_type)=>

min = 99999999.

max = 0.

idx = 1

ob\_threshold = ob\_filter == 'Atr' ? atr : cmean\_range

//Search for highest/lowest high within the structure interval and get range

if use\_max

for i = 1 to (n - loc)-1

if (high[i] - low[i]) < ob\_threshold[i] \* 2

max := math.max(high[i], max)

min := max == high[i] ? low[i] : min

idx := max == high[i] ? i : idx

else

for i = 1 to (n - loc)-1

if (high[i] - low[i]) < ob\_threshold[i] \* 2

min := math.min(low[i], min)

max := min == low[i] ? high[i] : max

idx := min == low[i] ? i : idx

array.unshift(target\_top, max)

array.unshift(target\_btm, min)

array.unshift(target\_left, time[idx])

array.unshift(target\_type, use\_max ? -1 : 1)

//Set order blocks

display\_ob(boxes, target\_top, target\_btm, target\_left, target\_type, show\_last, swing, size)=>

for i = 0 to math.min(show\_last-1, size-1)

get\_box = array.get(boxes, i)

box.set\_lefttop(get\_box, array.get(target\_left, i), array.get(target\_top, i))

box.set\_rightbottom(get\_box, array.get(target\_left, i), array.get(target\_btm, i))

box.set\_extend(get\_box, extend.right)

color css = na

if swing

if style == 'Monochrome'

css := array.get(target\_type, i) == 1 ? color.new(#b2b5be, 80) : color.new(#5d606b, 80)

border\_css = array.get(target\_type, i) == 1 ? #b2b5be : #5d606b

box.set\_border\_color(get\_box, border\_css)

else

css := array.get(target\_type, i) == 1 ? bull\_ob\_css : bear\_ob\_css

box.set\_border\_color(get\_box, css)

box.set\_bgcolor(get\_box, css)

else

if style == 'Monochrome'

css := array.get(target\_type, i) == 1 ? color.new(#b2b5be, 80) : color.new(#5d606b, 80)

else

css := array.get(target\_type, i) == 1 ? ibull\_ob\_css : ibear\_ob\_css

box.set\_border\_color(get\_box, css)

box.set\_bgcolor(get\_box, css)

//Line Style function

get\_line\_style(style) =>

out = switch style

'⎯⎯⎯' => line.style\_solid

'----' => line.style\_dashed

'····' => line.style\_dotted

//Set line/labels function for previous high/lows

phl(h, l, tf, css)=>

var line high\_line = line.new(na,na,na,na

, xloc = xloc.bar\_time

, color = css

, style = get\_line\_style(pdhl\_style))

var label high\_lbl = label.new(na,na

, xloc = xloc.bar\_time

, text = str.format('P{0}H', tf)

, color = TRANSP\_CSS

, textcolor = css

, size = size.small

, style = label.style\_label\_left)

var line low\_line = line.new(na,na,na,na

, xloc = xloc.bar\_time

, color = css

, style = get\_line\_style(pdhl\_style))

var label low\_lbl = label.new(na,na

, xloc = xloc.bar\_time

, text = str.format('P{0}L', tf)

, color = TRANSP\_CSS

, textcolor = css

, size = size.small

, style = label.style\_label\_left)

hy = ta.valuewhen(h != h[1], h, 1)

hx = ta.valuewhen(h == high, time, 1)

ly = ta.valuewhen(l != l[1], l, 1)

lx = ta.valuewhen(l == low, time, 1)

if barstate.islast

ext = time + (time - time[1])\*20

//High

line.set\_xy1(high\_line, hx, hy)

line.set\_xy2(high\_line, ext, hy)

label.set\_xy(high\_lbl, ext, hy)

//Low

line.set\_xy1(low\_line, lx, ly)

line.set\_xy2(low\_line, ext, ly)

label.set\_xy(low\_lbl, ext, ly)

//-----------------------------------------------------------------------------}

//Global variables

//-----------------------------------------------------------------------------{

var trend = 0, var itrend = 0

var top\_y = 0., var top\_x = 0

var btm\_y = 0., var btm\_x = 0

var itop\_y = 0., var itop\_x = 0

var ibtm\_y = 0., var ibtm\_x = 0

var trail\_up = high, var trail\_dn = low

var trail\_up\_x = 0, var trail\_dn\_x = 0

var top\_cross = true, var btm\_cross = true

var itop\_cross = true, var ibtm\_cross = true

var txt\_top = '', var txt\_btm = ''

//Alerts

bull\_choch\_alert = false

bull\_bos\_alert = false

bear\_choch\_alert = false

bear\_bos\_alert = false

bull\_ichoch\_alert = false

bull\_ibos\_alert = false

bear\_ichoch\_alert = false

bear\_ibos\_alert = false

bull\_iob\_break = false

bear\_iob\_break = false

bull\_ob\_break = false

bear\_ob\_break = false

eqh\_alert = false

eql\_alert = false

//Structure colors

var bull\_css = style == 'Monochrome' ? #b2b5be

: swing\_bull\_css

var bear\_css = style == 'Monochrome' ? #b2b5be

: swing\_bear\_css

var ibull\_css = style == 'Monochrome' ? #b2b5be

: swing\_ibull\_css

var ibear\_css = style == 'Monochrome' ? #b2b5be

: swing\_ibear\_css

//Labels size

var internal\_structure\_lbl\_size = internal\_structure\_size == 'Tiny'

? size.tiny

: internal\_structure\_size == 'Small'

? size.small

: size.normal

var swing\_structure\_lbl\_size = swing\_structure\_size == 'Tiny'

? size.tiny

: swing\_structure\_size == 'Small'

? size.small

: size.normal

var eqhl\_lbl\_size = eq\_size == 'Tiny'

? size.tiny

: eq\_size == 'Small'

? size.small

: size.normal

//Swings

[top, btm] = swings(length)

[itop, ibtm] = swings(5)

//-----------------------------------------------------------------------------}

//Pivot High

//-----------------------------------------------------------------------------{

var line extend\_top = na

var label extend\_top\_lbl = label.new(na, na

, color = TRANSP\_CSS

, textcolor = bear\_css

, style = label.style\_label\_down

, size = size.tiny)

if top

top\_cross := true

txt\_top := top > top\_y ? 'HH' : 'LH'

if show\_swings

top\_lbl = label.new(n-length, top, txt\_top

, color = TRANSP\_CSS

, textcolor = bear\_css

, style = label.style\_label\_down

, size = swing\_structure\_lbl\_size)

if mode == 'Present'

label.delete(top\_lbl[1])

//Extend recent top to last bar

line.delete(extend\_top[1])

extend\_top := line.new(n-length, top, n, top

, color = bear\_css)

top\_y := top

top\_x := n - length

trail\_up := top

trail\_up\_x := n - length

if itop

itop\_cross := true

itop\_y := itop

itop\_x := n - 5

//Trailing maximum

trail\_up := math.max(high, trail\_up)

trail\_up\_x := trail\_up == high ? n : trail\_up\_x

//Set top extension label/line

if barstate.islast and show\_hl\_swings

line.set\_xy1(extend\_top, trail\_up\_x, trail\_up)

line.set\_xy2(extend\_top, n + 20, trail\_up)

label.set\_x(extend\_top\_lbl, n + 20)

label.set\_y(extend\_top\_lbl, trail\_up)

label.set\_text(extend\_top\_lbl, trend < 0 ? 'Strong High' : 'Weak High')

//-----------------------------------------------------------------------------}

//Pivot Low

//-----------------------------------------------------------------------------{

var line extend\_btm = na

var label extend\_btm\_lbl = label.new(na, na

, color = TRANSP\_CSS

, textcolor = bull\_css

, style = label.style\_label\_up

, size = size.tiny)

if btm

btm\_cross := true

txt\_btm := btm < btm\_y ? 'LL' : 'HL'

if show\_swings

btm\_lbl = label.new(n - length, btm, txt\_btm

, color = TRANSP\_CSS

, textcolor = bull\_css

, style = label.style\_label\_up

, size = swing\_structure\_lbl\_size)

if mode == 'Present'

label.delete(btm\_lbl[1])

//Extend recent btm to last bar

line.delete(extend\_btm[1])

extend\_btm := line.new(n - length, btm, n, btm

, color = bull\_css)

btm\_y := btm

btm\_x := n-length

trail\_dn := btm

trail\_dn\_x := n-length

if ibtm

ibtm\_cross := true

ibtm\_y := ibtm

ibtm\_x := n - 5

//Trailing minimum

trail\_dn := math.min(low, trail\_dn)

trail\_dn\_x := trail\_dn == low ? n : trail\_dn\_x

//Set btm extension label/line

if barstate.islast and show\_hl\_swings

line.set\_xy1(extend\_btm, trail\_dn\_x, trail\_dn)

line.set\_xy2(extend\_btm, n + 20, trail\_dn)

label.set\_x(extend\_btm\_lbl, n + 20)

label.set\_y(extend\_btm\_lbl, trail\_dn)

label.set\_text(extend\_btm\_lbl, trend > 0 ? 'Strong Low' : 'Weak Low')

//-----------------------------------------------------------------------------}

//Order Blocks Arrays

//-----------------------------------------------------------------------------{

var iob\_top = array.new\_float(0)

var iob\_btm = array.new\_float(0)

var iob\_left = array.new\_int(0)

var iob\_type = array.new\_int(0)

var ob\_top = array.new\_float(0)

var ob\_btm = array.new\_float(0)

var ob\_left = array.new\_int(0)

var ob\_type = array.new\_int(0)

//-----------------------------------------------------------------------------}

//Pivot High BOS/CHoCH

//-----------------------------------------------------------------------------{

//Filtering

var bull\_concordant = true

if ifilter\_confluence

bull\_concordant := high - math.max(close, open) > math.min(close, open - low)

//Detect internal bullish Structure

if ta.crossover(close, itop\_y) and itop\_cross and top\_y != itop\_y and bull\_concordant

bool choch = na

if itrend < 0

choch := true

bull\_ichoch\_alert := true

else

bull\_ibos\_alert := true

txt = choch ? 'CHoCH' : 'BOS'

if show\_internals

if show\_ibull == 'All' or (show\_ibull == 'BOS' and not choch) or (show\_ibull == 'CHoCH' and choch)

display\_Structure(itop\_x, itop\_y, txt, ibull\_css, true, true, internal\_structure\_lbl\_size)

itop\_cross := false

itrend := 1

//Internal Order Block

if show\_iob

ob\_coord(false, itop\_x, iob\_top, iob\_btm, iob\_left, iob\_type)

//Detect bullish Structure

if ta.crossover(close, top\_y) and top\_cross

bool choch = na

if trend < 0

choch := true

bull\_choch\_alert := true

else

bull\_bos\_alert := true

txt = choch ? 'CHoCH' : 'BOS'

if show\_Structure

if show\_bull == 'All' or (show\_bull == 'BOS' and not choch) or (show\_bull == 'CHoCH' and choch)

display\_Structure(top\_x, top\_y, txt, bull\_css, false, true, swing\_structure\_lbl\_size)

//Order Block

if show\_ob

ob\_coord(false, top\_x, ob\_top, ob\_btm, ob\_left, ob\_type)

top\_cross := false

trend := 1

//-----------------------------------------------------------------------------}

//Pivot Low BOS/CHoCH

//-----------------------------------------------------------------------------{

var bear\_concordant = true

if ifilter\_confluence

bear\_concordant := high - math.max(close, open) < math.min(close, open - low)

//Detect internal bearish Structure

if ta.crossunder(close, ibtm\_y) and ibtm\_cross and btm\_y != ibtm\_y and bear\_concordant

bool choch = false

if itrend > 0

choch := true

bear\_ichoch\_alert := true

else

bear\_ibos\_alert := true

txt = choch ? 'CHoCH' : 'BOS'

if show\_internals

if show\_ibear == 'All' or (show\_ibear == 'BOS' and not choch) or (show\_ibear == 'CHoCH' and choch)

display\_Structure(ibtm\_x, ibtm\_y, txt, ibear\_css, true, false, internal\_structure\_lbl\_size)

ibtm\_cross := false

itrend := -1

//Internal Order Block

if show\_iob

ob\_coord(true, ibtm\_x, iob\_top, iob\_btm, iob\_left, iob\_type)

//Detect bearish Structure

if ta.crossunder(close, btm\_y) and btm\_cross

bool choch = na

if trend > 0

choch := true

bear\_choch\_alert := true

else

bear\_bos\_alert := true

txt = choch ? 'CHoCH' : 'BOS'

if show\_Structure

if show\_bear == 'All' or (show\_bear == 'BOS' and not choch) or (show\_bear == 'CHoCH' and choch)

display\_Structure(btm\_x, btm\_y, txt, bear\_css, false, false, swing\_structure\_lbl\_size)

//Order Block

if show\_ob

ob\_coord(true, btm\_x, ob\_top, ob\_btm, ob\_left, ob\_type)

btm\_cross := false

trend := -1

//-----------------------------------------------------------------------------}

//Order Blocks

//-----------------------------------------------------------------------------{

//Set order blocks

var iob\_boxes = array.new\_box(0)

var ob\_boxes = array.new\_box(0)

//Delete internal order blocks box coordinates if top/bottom is broken

for element in iob\_type

index = array.indexof(iob\_type, element)

if close < array.get(iob\_btm, index) and element == 1

array.remove(iob\_top, index)

array.remove(iob\_btm, index)

array.remove(iob\_left, index)

array.remove(iob\_type, index)

bull\_iob\_break := true

else if close > array.get(iob\_top, index) and element == -1

array.remove(iob\_top, index)

array.remove(iob\_btm, index)

array.remove(iob\_left, index)

array.remove(iob\_type, index)

bear\_iob\_break := true

//Delete internal order blocks box coordinates if top/bottom is broken

for element in ob\_type

index = array.indexof(ob\_type, element)

if close < array.get(ob\_btm, index) and element == 1

array.remove(ob\_top, index)

array.remove(ob\_btm, index)

array.remove(ob\_left, index)

array.remove(ob\_type, index)

bull\_ob\_break := true

else if close > array.get(ob\_top, index) and element == -1

array.remove(ob\_top, index)

array.remove(ob\_btm, index)

array.remove(ob\_left, index)

array.remove(ob\_type, index)

bear\_ob\_break := true

iob\_size = array.size(iob\_type)

ob\_size = array.size(ob\_type)

if barstate.isfirst

if show\_iob

for i = 0 to iob\_showlast-1

array.push(iob\_boxes, box.new(na,na,na,na, xloc = xloc.bar\_time))

if show\_ob

for i = 0 to ob\_showlast-1

array.push(ob\_boxes, box.new(na,na,na,na, xloc = xloc.bar\_time))

if iob\_size > 0

if barstate.islast

display\_ob(iob\_boxes, iob\_top, iob\_btm, iob\_left, iob\_type, iob\_showlast, false, iob\_size)

if ob\_size > 0

if barstate.islast

display\_ob(ob\_boxes, ob\_top, ob\_btm, ob\_left, ob\_type, ob\_showlast, true, ob\_size)

//-----------------------------------------------------------------------------}

//EQH/EQL

//-----------------------------------------------------------------------------{

var eq\_prev\_top = 0.

var eq\_top\_x = 0

var eq\_prev\_btm = 0.

var eq\_btm\_x = 0

if show\_eq

eq\_top = ta.pivothigh(eq\_len, eq\_len)

eq\_btm = ta.pivotlow(eq\_len, eq\_len)

if eq\_top

max = math.max(eq\_top, eq\_prev\_top)

min = math.min(eq\_top, eq\_prev\_top)

if max < min + atr \* eq\_threshold

eqh\_line = line.new(eq\_top\_x, eq\_prev\_top, n-eq\_len, eq\_top

, color = bear\_css

, style = line.style\_dotted)

eqh\_lbl = label.new(int(math.avg(n-eq\_len, eq\_top\_x)), eq\_top, 'EQH'

, color = #00000000

, textcolor = bear\_css

, style = label.style\_label\_down

, size = eqhl\_lbl\_size)

if mode == 'Present'

line.delete(eqh\_line[1])

label.delete(eqh\_lbl[1])

eqh\_alert := true

eq\_prev\_top := eq\_top

eq\_top\_x := n-eq\_len

if eq\_btm

max = math.max(eq\_btm, eq\_prev\_btm)

min = math.min(eq\_btm, eq\_prev\_btm)

if min > max - atr \* eq\_threshold

eql\_line = line.new(eq\_btm\_x, eq\_prev\_btm, n-eq\_len, eq\_btm

, color = bull\_css

, style = line.style\_dotted)

eql\_lbl = label.new(int(math.avg(n-eq\_len, eq\_btm\_x)), eq\_btm, 'EQL'

, color = #00000000

, textcolor = bull\_css

, style = label.style\_label\_up

, size = eqhl\_lbl\_size)

eql\_alert := true

if mode == 'Present'

line.delete(eql\_line[1])

label.delete(eql\_lbl[1])

eq\_prev\_btm := eq\_btm

eq\_btm\_x := n-eq\_len

//-----------------------------------------------------------------------------}

//Fair Value Gaps

//-----------------------------------------------------------------------------{

var bullish\_fvg\_max = array.new\_box(0)

var bullish\_fvg\_min = array.new\_box(0)

var bearish\_fvg\_max = array.new\_box(0)

var bearish\_fvg\_min = array.new\_box(0)

float bullish\_fvg\_avg = na

float bearish\_fvg\_avg = na

bullish\_fvg\_cnd = false

bearish\_fvg\_cnd = false

[src\_c1, src\_o1, src\_h, src\_l, src\_h2, src\_l2] =

request.security(syminfo.tickerid, fvg\_tf, get\_ohlc())

if show\_fvg

delta\_per = (src\_c1 - src\_o1) / src\_o1 \* 100

change\_tf = timeframe.change(fvg\_tf)

threshold = fvg\_auto ? ta.cum(math.abs(change\_tf ? delta\_per : 0)) / n \* 2

: 0

//FVG conditions

bullish\_fvg\_cnd := src\_l > src\_h2

and src\_c1 > src\_h2

and delta\_per > threshold

and change\_tf

bearish\_fvg\_cnd := src\_h < src\_l2

and src\_c1 < src\_l2

and -delta\_per > threshold

and change\_tf

//FVG Areas

if bullish\_fvg\_cnd

array.unshift(bullish\_fvg\_max, box.new(n-1, src\_l, n + fvg\_extend, math.avg(src\_l, src\_h2)

, border\_color = bull\_fvg\_css

, bgcolor = bull\_fvg\_css))

array.unshift(bullish\_fvg\_min, box.new(n-1, math.avg(src\_l, src\_h2), n + fvg\_extend, src\_h2

, border\_color = bull\_fvg\_css

, bgcolor = bull\_fvg\_css))

if bearish\_fvg\_cnd

array.unshift(bearish\_fvg\_max, box.new(n-1, src\_h, n + fvg\_extend, math.avg(src\_h, src\_l2)

, border\_color = bear\_fvg\_css

, bgcolor = bear\_fvg\_css))

array.unshift(bearish\_fvg\_min, box.new(n-1, math.avg(src\_h, src\_l2), n + fvg\_extend, src\_l2

, border\_color = bear\_fvg\_css

, bgcolor = bear\_fvg\_css))

for bx in bullish\_fvg\_min

if low < box.get\_bottom(bx)

box.delete(bx)

box.delete(array.get(bullish\_fvg\_max, array.indexof(bullish\_fvg\_min, bx)))

for bx in bearish\_fvg\_max

if high > box.get\_top(bx)

box.delete(bx)

box.delete(array.get(bearish\_fvg\_min, array.indexof(bearish\_fvg\_max, bx)))

//-----------------------------------------------------------------------------}

//Previous day/week high/lows

//-----------------------------------------------------------------------------{

//Daily high/low

[pdh, pdl] = request.security(syminfo.tickerid, 'D', hl()

, lookahead = barmerge.lookahead\_on)

//Weekly high/low

[pwh, pwl] = request.security(syminfo.tickerid, 'W', hl()

, lookahead = barmerge.lookahead\_on)

//Monthly high/low

[pmh, pml] = request.security(syminfo.tickerid, 'M', hl()

, lookahead = barmerge.lookahead\_on)

//Display Daily

if show\_pdhl

phl(pdh, pdl, 'D', pdhl\_css)

//Display Weekly

if show\_pwhl

phl(pwh, pwl, 'W', pwhl\_css)

//Display Monthly

if show\_pmhl

phl(pmh, pml, 'M', pmhl\_css)

//-----------------------------------------------------------------------------}

//Premium/Discount/Equilibrium zones

//-----------------------------------------------------------------------------{

var premium = box.new(na, na, na, na

, bgcolor = color.new(premium\_css, 80)

, border\_color = na)

var premium\_lbl = label.new(na, na

, text = 'Premium'

, color = TRANSP\_CSS

, textcolor = premium\_css

, style = label.style\_label\_down

, size = size.small)

var eq = box.new(na, na, na, na

, bgcolor = color.rgb(120, 123, 134, 80)

, border\_color = na)

var eq\_lbl = label.new(na, na

, text = 'Equilibrium'

, color = TRANSP\_CSS

, textcolor = eq\_css

, style = label.style\_label\_left

, size = size.small)

var discount = box.new(na, na, na, na

, bgcolor = color.new(discount\_css, 80)

, border\_color = na)

var discount\_lbl = label.new(na, na

, text = 'Discount'

, color = TRANSP\_CSS

, textcolor = discount\_css

, style = label.style\_label\_up

, size = size.small)

//Show Premium/Discount Areas

if barstate.islast and show\_sd

avg = math.avg(trail\_up, trail\_dn)

box.set\_lefttop(premium, math.max(top\_x, btm\_x), trail\_up)

box.set\_rightbottom(premium, n, .95 \* trail\_up + .05 \* trail\_dn)

label.set\_xy(premium\_lbl, int(math.avg(math.max(top\_x, btm\_x), n)), trail\_up)

box.set\_lefttop(eq, math.max(top\_x, btm\_x), .525 \* trail\_up + .475\*trail\_dn)

box.set\_rightbottom(eq, n, .525 \* trail\_dn + .475 \* trail\_up)

label.set\_xy(eq\_lbl, n, avg)

box.set\_lefttop(discount, math.max(top\_x, btm\_x), .95 \* trail\_dn + .05 \* trail\_up)

box.set\_rightbottom(discount, n, trail\_dn)

label.set\_xy(discount\_lbl, int(math.avg(math.max(top\_x, btm\_x), n)), trail\_dn)

//-----------------------------------------------------------------------------}

//Trend

//-----------------------------------------------------------------------------{

var color trend\_css = na

if show\_trend

if style == 'Colored'

trend\_css := itrend == 1 ? bull\_css : bear\_css

else if style == 'Monochrome'

trend\_css := itrend == 1 ? #b2b5be : #5d606b

plotcandle(open, high, low, close

, color = trend\_css

, wickcolor = trend\_css

, bordercolor = trend\_css

, editable = false)

//-----------------------------------------------------------------------------}

//Alerts

//-----------------------------------------------------------------------------{

//Internal Structure

alertcondition(bull\_ibos\_alert, 'Internal Bullish BOS', 'Internal Bullish BOS formed')

alertcondition(bull\_ichoch\_alert, 'Internal Bullish CHoCH', 'Internal Bullish CHoCH formed')

alertcondition(bear\_ibos\_alert, 'Internal Bearish BOS', 'Internal Bearish BOS formed')

alertcondition(bear\_ichoch\_alert, 'Internal Bearish CHoCH', 'Internal Bearish CHoCH formed')

//Swing Structure

alertcondition(bull\_bos\_alert, 'Bullish BOS', 'Internal Bullish BOS formed')

alertcondition(bull\_choch\_alert, 'Bullish CHoCH', 'Internal Bullish CHoCH formed')

alertcondition(bear\_bos\_alert, 'Bearish BOS', 'Bearish BOS formed')

alertcondition(bear\_choch\_alert, 'Bearish CHoCH', 'Bearish CHoCH formed')

//order Blocks

alertcondition(bull\_iob\_break, 'Bullish Internal OB Breakout', 'Price broke bullish internal OB')

alertcondition(bear\_iob\_break, 'Bearish Internal OB Breakout', 'Price broke bearish internal OB')

alertcondition(bull\_ob\_break, 'Bullish Swing OB Breakout', 'Price broke bullish swing OB')

alertcondition(bear\_ob\_break, 'Bearish Swing OB Breakout', 'Price broke bearish swing OB')

//EQH/EQL

alertcondition(eqh\_alert, 'Equal Highs', 'Equal highs detected')

alertcondition(eql\_alert, 'Equal Lows', 'Equal lows detected')

//FVG

alertcondition(bullish\_fvg\_cnd, 'Bullish FVG', 'Bullish FVG formed')

alertcondition(bearish\_fvg\_cnd, 'Bearish FVG', 'Bearish FVG formed')

//-----------------------------------------------------------------------------}