

```

//@version=5
//This source code is subject to the terms of the Mozilla Public License 2.0 at
https://mozilla.org/MPL/2.0/
indicator('OrderBlock / FVG / BoS Tools , overlay=true, max_boxes_count=500,
max_lines_count=500)

plotOB = input.bool(defval=true, title='Plot OB', group='Order Blocks')
obBullColor = input.color(defval=color.new(color.green, 90), title='Bullish OB
Color', inline='Set Custom Color', group='Order Blocks')
obBearColor = input.color(defval=color.new(color.red, 90), title='Bearish OB
Color', inline='Set Custom Color', group='Order Blocks')
obBoxBorder = input.string(defval=line.style_solid, title='OB Box Border Style',
options=[line.style_dashed, line.style_dotted, line.style_solid], group='Order
Blocks', tooltip='To disable border, set Border Width below to 0')
obBorderTransparency = input.int(defval=80, title='OB Border Box
Transparency', minval=0, maxval=100, group='Order Blocks')
obMaxBoxSet = input.int(defval=10, title='Maximum OB Box Displayed',
minval=1, maxval=100, group='Order Blocks', tooltip='Minimum = 1, Maximum =
100')
filterMitOB = input.bool(defval=false, title='Custom Color Mitigated OB',
group='Order Blocks')
mitOBColor = input.color(defval=color.new(color.gray, 90), title='Mitigated OB
Color', group='Order Blocks', inline='Set Custom Color Mit OB', tooltip='Set
Transparency to 0 to make mitigated OB disappear')

plotFVG = input.bool(defval=true, title='Plot FVG', group='Fair Value Gaps',
inline='FVG sets')
plotStructureBreakingFVG = input.bool(defval=true, title='Plot Structure
Breaking FVG', group='Fair Value Gaps', inline='FVG sets')
fvgBullColor = input.color(defval=color.new(color.black, 90), title='Bullish FVG
Color', inline='Set Custom Color', group='Fair Value Gaps')
fvgBearColor = input.color(defval=color.new(color.black, 90), title='Bearish
FVG Color', inline='Set Custom Color', group='Fair Value Gaps')
fvgStructBreakingColor = input.color(defval=color.new(color.blue, 90),
title='Structure Breaking FVG Color', inline='Set Custom Color', group='Fair
Value Gaps')
fvgBoxBorder = input.string(defval=line.style_solid, title='FVG Box Border
Style', options=[line.style_dashed, line.style_dotted, line.style_solid],
group='Fair Value Gaps', tooltip='To disable border, set Border Width below to
0')
fvgBorderTransparency = input.int(defval=80, title='FVG Border Box
Transparency', minval=0, maxval=100, group='Fair Value Gaps')
fvgMaxBoxSet = input.int(defval=10, title='Maximum FVG Box Displayed',
minval=1, maxval=100, group='Fair Value Gaps', tooltip='Minimum = 1, Maximum

```

```

= 100')
filterMitFVG = input.bool(defval=false, title='Custom Color Mitigated FVG',
group='Fair Value Gaps')
mitFVGColor = input.color(defval=color.new(color.gray, 90), title='Mitigated
FVG Color', group='Fair Value Gaps', inline='Set Custom Color Mit FVG',
tooltip='Set Transparency to 0 to make mitigated FVG disappear')

plotRJB = input.bool(defval=false, title='Plot RJB', group='Rejection Blocks',
inline='RJB sets')
rjbBullColor = input.color(defval=color.new(color.green, 90), title='Bullish RJB
Color', inline='Set Custom Color', group='Rejection Blocks')
rjbBearColor = input.color(defval=color.new(color.red, 90), title='Bearish RJB
Color', inline='Set Custom Color', group='Rejection Blocks')
rjbBoxBorder = input.string(defval=line.style_solid, title='RJB Box Border Style',
options=[line.style_dashed, line.style_dotted, line.style_solid], group='Rejection
Blocks', tooltip='To disable border, set Border Width below to 0')
rjbBorderTransparency = input.int(defval=80, title='RJB Border Box
Transparency', minval=0, maxval=100, group='Rejection Blocks')
rjbMaxBoxSet = input.int(defval=10, title='Maximum RJB Box Displayed',
minval=1, maxval=100, group='Rejection Blocks', tooltip='Minimum = 1,
Maximum = 100')
filterMitRJB = input.bool(defval=false, title='Custom Color Mitigated RJB',
group='Rejection Blocks')
mitRJBColor = input.color(defval=color.new(color.gray, 90), title='Mitigated RJB
Color', group='Rejection Blocks', inline='Set Custom Color Mit RJB', tooltip='Set
to 100 to make mitigated RJB disappear')

plotPVT = input.bool(defval=true, title='Plot Pivots', group='Pivots')
pivotLookup = input.int(defval=1, minval=1, maxval=5, title='Pivot Lookup',
group='Pivots', tooltip='Minimum = 1, Maximum = 5')
pvtTopColor = input.color(defval=color.new(color.silver, 0), title='Pivot Top
Color', group='Pivots', inline='PVT Color')
pvtBottomColor = input.color(defval=color.new(color.silver, 0), title='Pivot
Bottom Color', group='Pivots', inline='PVT Color')

plotBOS = input.bool(defval=false, title='Plot BoS', group='Crossovers',
inline='BOS sets')
useHighLowForBullishBoS = input.bool(defval=false, title='Use High/Low for
Bullish BoS (for Bearish setup)', group='Crossovers')
useHighLowForBearishBoS = input.bool(defval=false, title='Use High/Low for
Bearish BoS (for Bullish setup)', group='Crossovers')
bosBoxFlag = input.bool(title='BoS Box Length Manually', defval=false,
group='Crossovers', tooltip='If activated the BoS Boxes will not extend until
crossed by price. Instead will extend by the amount of bars choosen in the "Set
BoS Box Length Manually" option')
bosBoxLength = input.int(title='BoS Box Length Manually', defval=3, minval=1,

```

maxval=5, group='Crossovers', inline='BoS Boxes', tooltip='If "Set BoS Box Length Manually" is marked, choose by how many bars. Minimum = 1, Maximum = 5')

bosBullColor = input.color(defval=color.new(color.green, 90), title='Bullish BoS Color', inline='Set Custom Color', group='Crossovers')

bosBearColor = input.color(defval=color.new(color.red, 90), title='Bearish BoS Color', inline='Set Custom Color', group='Crossovers')

bosBoxBorder = input.string(defval=line.style_solid, title='BoS Box Border Style', options=[line.style_dashed, line.style_dotted, line.style_solid], group='Crossovers', tooltip='To disable border, set Border Width below to 0')

bosBorderTransparency = input.int(defval=80, title='BoS Border Box Transparency', minval=0, maxval=100, group='Crossovers')

bosMaxBoxSet = input.int(defval=10, title='Maximum BoS Box Displayed', minval=1, maxval=100, group='Crossovers', tooltip='Minimum = 1, Maximum = 100')

plotHVB = input.bool(defval=true, title='Plot HVB', group='High Volume Bar', tooltip='A candle where the average volume is higher than last few bars.')

hvbBullColor = input.color(defval=color.green, title='Bullish HVB Color', inline='Set Custom Color', group='High Volume Bar')

hvbBearColor = input.color(defval=color.red, title='Bearish HVB Color', inline='Set Custom Color', group='High Volume Bar')

hvbEMAPeriod = input.int(defval=12, minval=1, title='Volume EMA Period', group='High Volume Bar')

hvbMultiplier = input.float(defval=1.5, title='Volume Multiplier', minval=1, maxval=100, group='High Volume Bar')

plotPPDD = input.bool(defval=true, title="Plot PPDD OB's", group='Qualitative indicators', tooltip='Premium Premium Discount Discount (PPDD) is an OB formed after liquidity sweep. It will show up by default as a triangle (Bull ▲ / Bear ▼). Also PPDD1 (by default made with a x-cross x) which is a weak OB formed after liquidity sweep, that fails to completely engulf the high/low, but closes beyond the trapped candles open price.')

ppddBullColor = input.color(defval=color.new(color.green, 0), title="Bullish PPDD OB's Color", group='Qualitative indicators', inline='PPDD Color')

ppddBearColor = input.color(defval=color.new(color.red, 0), title="Bearish PPDD OB's Color", group='Qualitative indicators', inline='PPDD Color')

plotOBFVG = input.bool(defval=true, title='Plot Stacked OB+FVG', group='Qualitative indicators', tooltip='Marks the candle (default with a diamond ♦) when an OB & FVG are stacked, showing momentum')

obfvgBullColor = input.color(defval=color.new(color.green, 0), title='Bullish Stacked OB+FVG Color', group='Qualitative indicators', inline='OBFVG Color')

obfvgBearColor = input.color(defval=color.new(color.red, 0), title='Bearish Stacked OB+FVG Color', group='Qualitative indicators', inline='OBFVG Color')

```

plotLabelOB = input.bool(defval=true, title='Plot OB Label', inline='OB label',
group='Label Options')
obLabelColor = input.color(defval=color.gray, title='Color', inline='OB label',
group='Label Options')
obLabelSize = input.string(defval=size.tiny, title="Size", options=[size.huge,
size.large, size.small, size.tiny, size.auto, size.normal], inline='OB label',
group='Label Options')
plotLabelFVG = input.bool(defval=true, title='Plot FVG Label', inline='FVG label',
group='Label Options')
fvgLabelColor = input.color(defval=color.gray, title='Color', inline='FVG label',
group='Label Options')
fvgLabelSize = input.string(defval=size.tiny, title="Size", options=[size.huge,
size.large, size.small, size.tiny, size.auto, size.normal], inline='FVG label',
group='Label Options')
plotLabelRJB = input.bool(defval=true, title='Plot RJB Label', inline='RJB label',
group='Label Options')
rjbLabelColor = input.color(defval=color.gray, title='Color', inline='RJB label',
group='Label Options')
rjbLabelSize = input.string(defval=size.tiny, title="Size", options=[size.huge,
size.large, size.small, size.tiny, size.auto, size.normal], inline='RJB label',
group='Label Options')
plotLabelBOS = input.bool(defval=true, title='Plot BoS Label', inline='BOS label',
group='Label Options')
bosLabelColor = input.color(defval=color.gray, title='Color', inline='BOS label',
group='Label Options')
bosLabelSize = input.string(defval=size.tiny, title="Size", options=[size.huge,
size.large, size.small, size.tiny, size.auto, size.normal], inline='BOS label',
group='Label Options')

//Box Types
var int _ob = 1
var int _fvg = 2
var int _rjb = 3
var int _bos = 4

//Box Labels
var string _obLabel = "OB"
var string _fvgLabel = "FVG"
var string _rjbLabel = "RJB"
var string _bosLabel = "BoS"
var string _plus = "+"
var string _minus = "-"
var string _empty = ""

//Box Arrays
var box[] _bearBoxesOB = array.new_box()

```

```

var box[] _bullBoxesOB = array.new_box()
var box[] _bearBoxesFVG = array.new_box()
var box[] _bullBoxesFVG = array.new_box()
var box[] _bearBoxesRJB = array.new_box()
var box[] _bullBoxesRJB = array.new_box()
var box[] _bearBoxesBOS = array.new_box()
var box[] _bullBoxesBOS = array.new_box()

//Functions
isUp(index) =>
    close[index] > open[index]

isDown(index) =>
    close[index] < open[index]

isObUp(index) =>
    isDown(index + 1) and isUp(index) and close[index] > high[index + 1]

isObDown(index) =>
    isUp(index + 1) and isDown(index) and close[index] < low[index + 1]

isFvgUp(index) =>
    (low[index] > high[index + 2])

isFvgDown(index) =>
    (high[index] < low[index + 2])

//Function to Calculte Box Length
_controlBox(_boxes, _high, _low, _type) =>
    if array.size(_boxes) > 0
        for i = array.size(_boxes) - 1 to 0 by 1
            _box = array.get(_boxes, i)
            _boxLow = box.get_bottom(_box)
            _boxHigh = box.get_top(_box)
            _boxRight = box.get_right(_box)
            if bosBoxFlag and _type == _bos
                if na or (bar_index + bosBoxLength - 1 == _boxRight and not((_high >
                _boxLow and _low < _boxLow) or (_high > _boxHigh and _low < _boxHigh)))
                    box.set_right(_box, bar_index + bosBoxLength - 1)
                else if (filterMitOB and _type == _ob) or (filterMitFVG and _type == _fvg)
                or (filterMitRJB and _type == _rjb)
                    if na or (bar_index == _boxRight and not((_high > _boxLow and _low <
                    _boxLow) or (_high > _boxHigh and _low < _boxHigh)))
                        box.set_right(_box, bar_index + 1)
            else
                if _type == _ob

```

```

        box.set_bgcolor(_box, mitOBColor)
        box.set_border_color(_box, mitOBColor)
    else if _type == _fvg
        box.set_bgcolor(_box, mitFVGColor)
        box.set_border_color(_box, mitFVGColor)
    else if _type == _rjb
        box.set_bgcolor(_box, mitRJBColor)
        box.set_border_color(_box, mitRJBColor)
    else
        if na or (bar_index == _boxRight and not((_high > _boxLow and _low <
        _boxLow) or (_high > _boxHigh and _low < _boxHigh)))
            box.set_right(_box, bar_index + 1)

```

```

////////// Pivots //////////

```

```

hih = ta.pivohigh(high, pivotLookup, pivotLookup)
lol = ta.pivotlow(low, pivotLookup, pivotLookup)
top = ta.valuwhen(hih, high[pivotLookup], 0)
bottom = ta.valuwhen(lol, low[pivotLookup], 0)
plot(top, offset=-pivotLookup, linewidth=1, color=(top != top[1] ? na :
(plotPVT ? pvtTopColor : na)), title="Pivot Top")
plot(bottom, offset=-pivotLookup, linewidth=1, color=(bottom != bottom[1] ?
na : (plotPVT ? pvtBottomColor : na)), title="Pivot Bottom")

```

```

////////// Order Block //////////

```

```

//Bullish OB Box Plotting

```

```

if isObUp(1) and plotOB

```

```

    _bullboxOB = box.new(left=bar_index - 2, top=high[2], right=bar_index,
    bottom=math.min(low[2], low[1]), border_color=color.new(obBullColor,
    obBorderTransparency), border_style=obBoxBorder, border_width=1,
    bgcolor=obBullColor,
    text=plotLabelOB ? _obLabel + _plus : _empty, text_halign=text.align_right,
    text_valign=text.align_bottom, text_size=obLabelSize, text_color=obLabelColor)
    if array.size(_bullBoxesOB) > obMaxBoxSet
        box.delete(array.shift(_bullBoxesOB))
    array.push(_bullBoxesOB, _bullboxOB)

```

```

//Bearish OB Box Plotting

```

```

if isObDown(1) and plotOB

```

```

    _bearboxOB = box.new(left=bar_index - 2, top=math.max(high[2], high[1]),
    right=bar_index, bottom=low[2], border_color=color.new(obBearColor,
    obBorderTransparency), border_style=obBoxBorder, border_width=1,
    bgcolor=obBearColor,
    text=plotLabelOB ? _obLabel + _minus : _empty, text_halign=text.align_right,
    text_valign=text.align_bottom, text_size=obLabelSize, text_color=obLabelColor)
    if array.size(_bearBoxesOB) > obMaxBoxSet
        box.delete(array.shift(_bearBoxesOB))

```

```

array.push(_bearBoxesOB, _bearboxOB)

if plotOB
  _controlBox(_bearBoxesOB, high, low, _ob)
  _controlBox(_bullBoxesOB, high, low, _ob)

////////// Fair Value Gap //////////
//Bullish FVG Box Plotting
if isFvgUp(0)
  box _bullboxFVG = na
  if plotStructureBreakingFVG and (close[1] > top) and (low[1] < top) and
(high[2] < top) and (low > top)
    _bullboxFVG := box.new(left=bar_index-2, top=low[0], right=bar_index,
bottom=high[2], bgcolor=fvgStructBreakingColor,
border_color=color.new(fvgStructBreakingColor, fvgBorderTransparency),
border_style=fvgBoxBorder, border_width=1,
text=plotLabelFVG ? _fvgLabel + _plus : _empty,
text_halign=text.align_right, text_valign=text.align_bottom,
text_size=fvgLabelSize, text_color=fvgLabelColor)
  else if plotFVG
    _bullboxFVG := box.new(left=bar_index-2, top=low[0], right=bar_index,
bottom=high[2], bgcolor=fvgBullColor, border_color=color.new(fvgBullColor,
fvgBorderTransparency), border_style=fvgBoxBorder, border_width=1,
text=plotLabelFVG ? _fvgLabel + _plus : _empty,
text_halign=text.align_right, text_valign=text.align_bottom,
text_size=fvgLabelSize, text_color=fvgLabelColor)
  if array.size(_bullBoxesFVG) > fvgMaxBoxSet
    box.delete(array.shift(_bullBoxesFVG))
    array.push(_bullBoxesFVG, _bullboxFVG)

//Bearish FVG Box Plotting
if isFvgDown(0)
  box _bearboxFVG = na
  if plotStructureBreakingFVG and (close[1] < bottom) and (high[1] > bottom)
and (low[2] > bottom) and (high < bottom)
    _bearboxFVG := box.new(left=bar_index-2, top=low[2], right=bar_index,
bottom=high[0], bgcolor=fvgStructBreakingColor,
border_color=color.new(fvgStructBreakingColor, fvgBorderTransparency),
border_style=fvgBoxBorder, border_width=1,
text=plotLabelFVG ? _fvgLabel + _minus : _empty,
text_halign=text.align_right, text_valign=text.align_bottom,
text_size=fvgLabelSize, text_color=fvgLabelColor)
  else if plotFVG
    _bearboxFVG := box.new(left=bar_index-2, top=low[2], right=bar_index,
bottom=high[0], bgcolor=fvgBearColor, border_color=color.new(fvgBearColor,
fvgBorderTransparency), border_style=fvgBoxBorder, border_width=1,

```

```

    text=plotLabelFVG ? _fvgLabel + _minus : _empty,
text_halign=text.align_right, text_valign=text.align_bottom,
text_size=fvgLabelSize, text_color=fvgLabelColor)
    if array.size(_bearBoxesFVG) > fvgMaxBoxSet
        box.delete(array.shift(_bearBoxesFVG))
    array.push(_bearBoxesFVG, _bearboxFVG)

if plotFVG or plotStructureBreakingFVG
    _controlBox(_bearBoxesFVG, high, low, _fvg)
    _controlBox(_bullBoxesFVG, high, low, _fvg)

////////// Rejection Block //////////
if plotRJB
    isDownRjbObCondition = isObDown(1)
    isDownRjb1 = isDownRjbObCondition and (high[1] < (close[2] +
0.2*(high[2]-close[2]))) // RJB is on trapped's wick and <50% of the wick was
covered by signal
    isDownRjb2 = isDownRjbObCondition and (high[1] > high[2]) // RJB is on
signal's wick
    if isDownRjb1 and plotRJB
        _bearboxRJB = box.new(left=bar_index-2, top=high[2], right=bar_index,
bottom=close[2], bgcolor=rjbBearColor, border_color=color.new(rjbBearColor,
rjbBorderTransparency), border_style=rjbBoxBorder, border_width=1,
        text=plotLabelRJB ? _rjbLabel + _minus : _empty,
text_halign=text.align_right, text_valign=text.align_bottom,
text_size=rjbLabelSize, text_color=rjbLabelColor)
        if array.size(_bearBoxesRJB) > rjbMaxBoxSet
            box.delete(array.shift(_bearBoxesRJB))
        array.push(_bearBoxesRJB, _bearboxRJB)

    if isDownRjb2 and plotRJB
        _bearboxRJB = box.new(left=bar_index-1, top=high[1], right=bar_index,
bottom=open[1], bgcolor=rjbBearColor, border_color=color.new(rjbBearColor,
rjbBorderTransparency), border_style=rjbBoxBorder, border_width=1,
        text=plotLabelRJB ? _rjbLabel + _minus : _empty,
text_halign=text.align_right, text_valign=text.align_bottom,
text_size=rjbLabelSize, text_color=rjbLabelColor)
        if array.size(_bearBoxesRJB) > rjbMaxBoxSet
            box.delete(array.shift(_bearBoxesRJB))
        array.push(_bearBoxesRJB, _bearboxRJB)

//Bullish RJB Box Plotting
if plotRJB
    isUpRjbObCondition = isObUp(1)
    isUpRjb1 = isUpRjbObCondition and (low[1] > (close[2] - 0.2*(close[2]-
low[2]))) // RJB is on trapped's wick and <50% of the wick was covered by

```

```

signal
  isUpRjb2 = isUpRjbObCondition and (low[1] < low[2]) // RJB is on signal's
wick
  if isUpRjb1 and plotRJB
    _bullboxRJB = box.new(left=bar_index-2, top=close[2], right=bar_index,
bottom=low[2], bgcolor=rjbBullColor, border_color=color.new(rjbBullColor,
rjbBorderTransparency), border_style=rjbBoxBorder, border_width=1,
    text=plotLabelRJB ? _rjbLabel + _plus : _empty,
text_halign=text.align_right, text_valign=text.align_bottom,
text_size=rjbLabelSize, text_color=rjbLabelColor)
    if array.size(_bullBoxesRJB) > rjbMaxBoxSet
      box.delete(array.shift(_bullBoxesRJB))
    array.push(_bullBoxesRJB, _bullboxRJB)

  if isUpRjb2 and plotRJB
    _bullboxRJB = box.new(left=bar_index-1, top=open[1], right=bar_index,
bottom=low[1], bgcolor=rjbBullColor, border_color=color.new(rjbBullColor,
rjbBorderTransparency), border_style=rjbBoxBorder, border_width=1,
    text=plotLabelRJB ? _rjbLabel + _plus : _empty,
text_halign=text.align_right, text_valign=text.align_bottom,
text_size=rjbLabelSize, text_color=rjbLabelColor)
    if array.size(_bullBoxesRJB) > rjbMaxBoxSet
      box.delete(array.shift(_bullBoxesRJB))
    array.push(_bullBoxesRJB, _bullboxRJB)

if plotRJB
  _controlBox(_bearBoxesRJB, high, low, _rjb)
  _controlBox(_bullBoxesRJB, high, low, _rjb)

////////// Crossovers a.k.a. Break of Structure //////////
//Bullish BOS Box Plotting
if plotBOS
  if ta.crossover(useHighLowForBullishBoS ? high : close, top)
    _bullboxBOS = box.new(left=bar_index, top=top, right=bosBoxFlag ?
bar_index+bosBoxLength : bar_index+1, bottom=bottom, bgcolor=bosBullColor,
border_color=color.new(bosBullColor, bosBorderTransparency),
border_style=bosBoxBorder, border_width=1,
    text=plotLabelBOS ? _bosLabel + _plus : _empty,
text_halign=text.align_right, text_valign=text.align_bottom,
text_size=bosLabelSize, text_color=bosLabelColor)
    if array.size(_bullBoxesBOS) > bosMaxBoxSet
      box.delete(array.shift(_bullBoxesBOS))
    array.push(_bullBoxesBOS, _bullboxBOS)

//Bearish BOS Box Plotting
if plotBOS

```

```

    if ta.crossunder(useHighLowForBearishBoS ? low : close, bottom)
        _bearboxBOS = box.new(left=bar_index, top=top, right=bosBoxFlag ?
bar_index+bosBoxLength : bar_index+1, bottom=bottom,
bgcolor=bosBearColor, border_color=color.new(bosBearColor,
bosBorderTransparency), border_style=bosBoxBorder, border_width=1,
    text=plotLabelBOS ? _bosLabel + _minus : _empty,
text_halign=text.align_right, text_valign=text.align_bottom,
text_size=bosLabelSize, text_color=bosLabelColor)
        if array.size(_bearBoxesBOS) > bosMaxBoxSet
            box.delete(array.shift(_bearBoxesBOS))
            array.push(_bearBoxesBOS, _bearboxBOS)

```

```

if plotBOS
    _controlBox(_bearBoxesBOS, high, low, _bos)
    _controlBox(_bullBoxesBOS, high, low, _bos)

```

```

//////////////////// Premium Premium & Discount Discount //////////////////////
premiumPremium = plotPPDD and isObDown(0) and ((math.max(high, high[1])
> top and close < top) or (math.max(high, high[1]) > top[1] and close < top[1]))
discountDiscount = plotPPDD and isObUp(0) and ((math.min(low, low[1]) <
bottom and close > bottom) or (math.min(low, low[1]) < bottom[1] and close >
bottom[1]))
plotshape(premiumPremium, "Bearish PPDD OB", style=shape.triangledown ,
location=location.abovebar, color=ppddBearColor, size=size.tiny)
plotshape(discountDiscount, "Bullish PPDD OB", style=shape.triangleup ,
location=location.belowbar, color=ppddBullColor, size=size.tiny)

```

```

premiumPremium1 = plotPPDD and (isUp(1) and isDown(0) and close[0] <
open[1]) and ((math.max(high, high[1]) > top and close < top) or
(math.max(high, high[1]) > top[1] and close < top[1])) and not
premiumPremium
discountDiscount1 = plotPPDD and (isDown(1) and isUp(0) and close[0] >
open[1]) and ((math.min(low, low[1]) < bottom and close > bottom) or
(math.min(low, low[1]) < bottom[1] and close > bottom[1])) and not
discountDiscount
plotshape(premiumPremium1, "Bearish PPDD Weak OB", style=shape.xcross,
location=location.abovebar, color=ppddBearColor, size=size.tiny)
plotshape(discountDiscount1, "Bullish PPDD Weak OB", style=shape.xcross,
location=location.belowbar, color=ppddBullColor, size=size.tiny)

```

```

//////////////////// High Volume Bars //////////////////////
volEma = ta.ema(volume, hvbEMAPeriod)
isHighVolume = volume > (hvbMultiplier * volEma)
barcolor(plotHVB and isUp(0) and isHighVolume ? hvbBullColor : na,
title="Bullish HVB")
barcolor(plotHVB and isDown(0) and isHighVolume ? hvbBearColor : na,

```

title="Bearish HVB")

////////// Stacked OB + FVG //////////

plotshape(plotOBFVG and isFvgDown(0) and isObDown(1), "Bearish OB+FVG Stack", style=shape.diamond, location=location.abovebar, color=obfvgBearColor, size=size.tiny)

plotshape(plotOBFVG and isFvgUp(0) and isObUp(1), "Bullish OB+FVG Stack", style=shape.diamond, location=location.belowbar, color=obfvgBullColor, size=size.tiny)