```
// This Pine Script® code is subject to the terms of the Mozilla Public License 2.0 at https://mozilla.org/MPL/2.0/

//@version=6

indicator("MFB FVG-123 Delta Story Labels", shorttitle = "FVG-123 Delta Story", overlay = true, max_labels_count = 500)


// --- Inputs ---

int footprintTicksInput = input.int(4, "Footprint ticks per row", minval = 1, maxval = 100, tooltip = "Price grouping used to read real-trade footprint delta on each chart bar.", group = "Order Flow")

int approachBarsInput = input.int(4, "Completed approach bars", minval = 3, maxval = 5, tooltip = "Number of completed bars immediately before the first FVG touch used for approach delta and price-efficiency context.", group = "Order Flow")

float minDepartureDeltaFractionInput = input.float(0.10, "Minimum confirmation delta / volume", minval = 0.01, maxval = 0.95, step = 0.01, tooltip = "A flow pass requires the FVG-123 confirmation candle's delta, as a fraction of its footprint volume, to meet this threshold in the setup direction. Approach and interaction readings are shown as context.", group = "Qualification")

color bullColorInput = input.color(#089981, "Bullish label", tooltip = "Label color for bullish FVG-123 events.", group = "Style")

color bearColorInput = input.color(#f23645, "Bearish label", tooltip = "Label color for bearish FVG-123 events.", group = "Style")

color otherColorInput = input.color(#5b9cf6, "Not qualified / data unavailable", tooltip = "Label color when the structure confirms but flow does not pass or required footprint readings are missing.", group = "Style")
```

```
string labelSizeInput = input.string("Normal", "Label text size", options =
["Tiny", "Small", "Normal", "Large", "Huge"], tooltip = "Adjust the size of the
simple label text.", group = "Style")


// --- Formatting helpers ---
f_num(float value) =>
    na(value) ? "n/a" : str.tostring(value, format.volume)


f_pct(float value) =>
    na(value) ? "n/a" : str.tostring(value * 100.0, "#.0") + "%"


f_label_size(string sizeName) =>
    switch sizeName
        "Tiny" => size.tiny
        "Small" => size.small
        "Normal" => size.normal
        "Large" => size.large
        => size.huge


labelSize = f_label_size(labelSizeInput)


// --- Footprint and approach measurements ---
footprint footprintData = request.footprint(footprintTicksInput)
float barDelta = na(footprintData) ? na : footprint.delta(footprintData)
```

```
float barVolume = na(footprintData) ? na :
footprint.total_volume(footprintData)

bool barFlowValid = not na(barDelta) and not na(barVolume) and barVolume >
0.0

float approachCoverage = ta.sma(barFlowValid ? 1.0 : 0.0,
approachBarsInput)[1]

float approachDeltaSum = ta.sma(nz(barDelta, 0.0), approachBarsInput)[1] *
approachBarsInput

float approachVolumeSum = ta.sma(nz(barVolume, 0.0),
approachBarsInput)[1] * approachBarsInput

float approachDeltaFraction = not na(approachVolumeSum) and
approachVolumeSum > 0.0 ? approachDeltaSum / approachVolumeSum : na

float approachTravel = ta.sma(math.abs(close - close[1]),
approachBarsInput)[1] * approachBarsInput

float approachNetMove = close[1] - close[approachBarsInput + 1]

float approachEfficiency = not na(approachTravel) and approachTravel > 0.0 ?
math.abs(approachNetMove) / approachTravel : na

bool approachDataValid = not na(approachCoverage) and approachCoverage
>= 0.999 and not na(approachVolumeSum) and approachVolumeSum > 0.0

float departureDeltaFraction = barFlowValid ? barDelta / barVolume : na


// --- Latest active bullish and bearish FVG-123 candidates ---

var bool bullActive = false

var float bullTop = na

var float bullBottom = na

var int bullCreatedBar = na
```

```
var bool bullTouched = false
var int bullTouchBar = na
var float bullInteractionDelta = 0.0
var float bullInteractionVolume = 0.0
var bool bullInteractionValid = false
var float bullMaxPenetration = 0.0
var float bullApproachDelta = na
var float bullApproachFraction = na
var float bullApproachEfficiency = na
var bool bullApproachValid = false

var bool bearActive = false
var float bearTop = na
var float bearBottom = na
var int bearCreatedBar = na
var bool bearTouched = false
var int bearTouchBar = na
var float bearInteractionDelta = 0.0
var float bearInteractionVolume = 0.0
var bool bearInteractionValid = false
var float bearMaxPenetration = 0.0
var float bearApproachDelta = na
var float bearApproachFraction = na
```

```
var float bearApproachEfficiency = na
var bool bearApproachValid = false

bool bullishFvg123DeltaSignal = false
bool bearishFvg123DeltaSignal = false

// Process only confirmed bars. A candidate is invalidated when its FVG is fully filled.
if barstate.isconfirmed and bullActive
   if low <= bullBottom
      bullActive := false
   else if bullTouched and bar_index > bullTouchBar and close > bullTop
      float bullDepartureFraction = departureDeltaFraction
      bool bullStoryDataValid = bullApproachValid and bullInteractionValid and barFlowValid
      bool bullFlowPass = bullStoryDataValid and bullDepartureFraction >= minDepartureDeltaFractionInput
      bullishFvg123DeltaSignal := bullFlowPass
      string bullStatus = not bullStoryDataValid ? "DATA UNAVAILABLE" : bullFlowPass ? "FLOW PASS" : "FLOW NOT QUALIFIED"
      float bullInteractionFraction = bullInteractionVolume > 0.0 ? bullInteractionDelta / bullInteractionVolume : na
      float bullPenetrationFraction = bullTop > bullBottom ? math.min(bullMaxPenetration / (bullTop - bullBottom), 1.0) : na
```

```
    string bullText = bullFlowPass ? "Bullish FVG-123 delta" : not
bullStoryDataValid ? "Bullish FVG-123 delta\nData unavailable" : "Bullish FVG-
123 delta\nNot qualified"

    color bullLabelColor = bullFlowPass ? bullColorInput : otherColorInput

    label.new(x = math.max(0, bullCreatedBar - 2), y = math.avg(bullTop,
bullBottom), text = bullText, xloc = xloc.bar_index, yloc = yloc.price, color =
color.new(bullLabelColor, 10), style = label.style_label_right, textcolor =
color.white, size = labelSize)

    bullActive := false

  else if not bullTouched and bar_index > bullCreatedBar and high >=
bullBottom and low <= bullTop

    bullTouched := true

    bullTouchBar := bar_index

    bullInteractionDelta := nz(barDelta, 0.0)

    bullInteractionVolume := nz(barVolume, 0.0)

    bullInteractionValid := barFlowValid

    bullMaxPenetration := math.min(math.max(bullTop - low, 0.0), bullTop -
bullBottom)

    bullApproachDelta := approachDeltaSum

    bullApproachFraction := approachDeltaFraction

    bullApproachEfficiency := approachEfficiency

    bullApproachValid := approachDataValid

  else if bullTouched

    bullInteractionDelta += nz(barDelta, 0.0)

    bullInteractionVolume += nz(barVolume, 0.0)
```

```
        bullInteractionValid := bullInteractionValid and barFlowValid
        bullMaxPenetration := math.max(bullMaxPenetration,
math.min(math.max(bullTop - low, 0.0), bullTop - bullBottom))


if barstate.isconfirmed and bearActive
    if high >= bearTop
        bearActive := false
    else if bearTouched and bar_index > bearTouchBar and close < bearBottom
        float bearDepartureFraction = departureDeltaFraction
        bool bearStoryDataValid = bearApproachValid and bearInteractionValid
and barFlowValid
        bool bearFlowPass = bearStoryDataValid and bearDepartureFraction <= -
minDepartureDeltaFractionInput
        bearishFvg123DeltaSignal := bearFlowPass
        string bearStatus = not bearStoryDataValid ? "DATA UNAVAILABLE" :
bearFlowPass ? "FLOW PASS" : "FLOW NOT QUALIFIED"
        float bearInteractionFraction = bearInteractionVolume > 0.0 ?
bearInteractionDelta / bearInteractionVolume : na
        float bearPenetrationFraction = bearTop > bearBottom ?
math.min(bearMaxPenetration / (bearTop - bearBottom), 1.0) : na
        string bearText = bearFlowPass ? "Bearish FVG-123 delta" : not
bearStoryDataValid ? "Bearish FVG-123 delta\nData unavailable" : "Bearish
FVG-123 delta\nNot qualified"
        color bearLabelColor = bearFlowPass ? bearColorInput : otherColorInput
        label.new(x = math.max(0, bearCreatedBar - 2), y = math.avg(bearTop,
bearBottom), text = bearText, xloc = xloc.bar_index, yloc = yloc.price, color =
```

```
color.new(bearLabelColor, 10), style = label.style_label_right, textcolor =
color.white, size = labelSize)
        bearActive := false
    else if not bearTouched and bar_index > bearCreatedBar and high >=
bearBottom and low <= bearTop
        bearTouched := true
        bearTouchBar := bar_index
        bearInteractionDelta := nz(barDelta, 0.0)
        bearInteractionVolume := nz(barVolume, 0.0)
        bearInteractionValid := barFlowValid
        bearMaxPenetration := math.min(math.max(high - bearBottom, 0.0),
bearTop - bearBottom)
        bearApproachDelta := approachDeltaSum
        bearApproachFraction := approachDeltaFraction
        bearApproachEfficiency := approachEfficiency
        bearApproachValid := approachDataValid
    else if bearTouched
        bearInteractionDelta += nz(barDelta, 0.0)
        bearInteractionVolume += nz(barVolume, 0.0)
        bearInteractionValid := bearInteractionValid and barFlowValid
        bearMaxPenetration := math.max(bearMaxPenetration,
math.min(math.max(high - bearBottom, 0.0), bearTop - bearBottom))

// Start new three-candle FVG candidates after processing older candidates.
```

```
if barstate.isconfirmed and bar_index >= 2 and low > high[2]
    bullTop := low
    bullBottom := high[2]
    bullCreatedBar := bar_index
    bullTouched := false
    bullTouchBar := na
    bullInteractionDelta := 0.0
    bullInteractionVolume := 0.0
    bullInteractionValid := false
    bullMaxPenetration := 0.0
    bullApproachDelta := na
    bullApproachFraction := na
    bullApproachEfficiency := na
    bullApproachValid := false
    bullActive := true

if barstate.isconfirmed and bar_index >= 2 and high < low[2]
    bearTop := low[2]
    bearBottom := high
    bearCreatedBar := bar_index
    bearTouched := false
    bearTouchBar := na
    bearInteractionDelta := 0.0
```

```
    bearInteractionVolume := 0.0
    bearInteractionValid := false
    bearMaxPenetration := 0.0
    bearApproachDelta := na
    bearApproachFraction := na
    bearApproachEfficiency := na
    bearApproachValid := false
    bearActive := true


// --- Alerts ---
bool anyFvg123DeltaSignal = bullishFvg123DeltaSignal or
bearishFvg123DeltaSignal
alertcondition(bullishFvg123DeltaSignal, title = "Bullish FVG-123 delta",
message = "Bullish FVG-123 delta on {{ticker}} ({{interval}}).")
alertcondition(bearishFvg123DeltaSignal, title = "Bearish FVG-123 delta",
message = "Bearish FVG-123 delta on {{ticker}} ({{interval}}).")
alertcondition(anyFvg123DeltaSignal, title = "Any FVG-123 delta", message =
"FVG-123 delta signal on {{ticker}} ({{interval}}).")
```