

// 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 Footprints with EMAs in Agreement', overlay = true, max\_labels\_count = 500,  
max\_boxes\_count = 500, max\_lines\_count = 500)

// --- Colors ---

COLOR\_BULL = #089981

COLOR\_BEAR = #f23645

COLOR\_ENTRY = #5b9cf6

COLOR\_FLOW = #00bcd4 // Cyan/Light Blue for arrows

// --- Static Alert Declaration ---

var bool global\_alert\_triggered = false

alertcondition(global\_alert\_triggered, title = "Footprints - Flow/Sweep", message = "A high-  
probability Flow or Sweep footprint signal (EMA Aligned) has been confirmed.")

// --- Inputs ---

grpHTF = '🌐 HTF Filter Settings'

use\_15m = input.bool(true, 'Include 15m FVG', group = grpHTF)

use\_1h = input.bool(true, 'Include 1h FVG', group = grpHTF)

use\_4h = input.bool(true, 'Include 4h FVG', group = grpHTF)

grpLiq = '⚡ Liquidity Filter'

touch\_lookback = input.int(50, 'Liquidity Touch Lookback (Bars)', minval = 1, group = grpLiq)

grpEMA = '📊 MTF EMA Filter (20 Period)'

use\_ema\_filter = input.bool(true, 'Use 1m/5m/15m EMA Alignment', group = grpEMA)

show\_emas = input.bool(true, 'Show EMAs on Chart', group = grpEMA)

grpAir = ' 🚀 Air Support Settings'

slope\_thresh = input.float(20.0, 'Min 15m EMA Slope (Degrees)', minval = 0, step = 1, group = grpAir, tooltip = "The minimum angle of the 15m EMA required for 'Air Support'")

dist\_thresh = input.float(15.0, 'Min 1m-15m Dist (Points)', minval = 0, step = 1, group = grpAir, tooltip = "The minimum vertical distance between the 1m EMA and 15m EMA in price points.")

grpTrade = ' 💰 Trade Management'

sl\_mode = input.string('FVG Midpoint', 'Stop Loss Mode', options = ['FVG Edge', 'FVG Midpoint', 'Signal Candle'], group = grpTrade, tooltip = "FVG Edge: Traditional SL at the far side of the gap. \nFVG Midpoint: Tighter SL at the 50% level of the gap. \nSignal Candle: Aggressive SL at the low/high of the entry candle.")

rr\_mult = input.float(2.0, 'Risk:Reward Ratio', minval = 1.0, step = 0.5, group = grpTrade, tooltip = "The target Reward relative to the Risk (e.g., 2.0 means target twice the risk).")

min\_rr = input.float(1.0, 'Min RR (Liquidity Cap)', minval = 0.5, step = 0.1, group = grpTrade, tooltip = "Prevents the TP from being moved too close to the entry when targeting PDH/PWH levels.")

use\_be = input.bool(true, 'Use Break-Even SL', group = grpTrade, tooltip = "Automatically moves the SL to the entry price once a certain profit is reached.")

be\_trigger\_rr = input.float(1.0, 'BE Trigger (RR)', minval = 0.1, step = 0.1, group = grpTrade, tooltip = "Moves SL to BE once price reaches this Reward-to-Risk ratio (e.g., 1.0 = Move to BE at 1:1 profit).")

grpVisual = ' 🎨 Visual Settings'

show\_levels = input.bool(true, 'Show Session H/L Levels', group = grpVisual)

lookback\_hrs = input.int(48, 'Show Trades from Last (Hours)', minval = 1, group = grpVisual)

```
line_len = input.int(15, 'Management Line Length', minval = 5, group = grpVisual)
```

```
// --- HTF Data & Levels ---
```

```
pdh = request.security(syminfo.tickerid, 'D', high[1], lookahead = barmerge.lookahead_on)
```

```
pdl = request.security(syminfo.tickerid, 'D', low[1], lookahead = barmerge.lookahead_on)
```

```
pwh = request.security(syminfo.tickerid, 'W', high[1], lookahead = barmerge.lookahead_on)
```

```
pwl = request.security(syminfo.tickerid, 'W', low[1], lookahead = barmerge.lookahead_on)
```

```
// MTF EMAs
```

```
ema1 = request.security(syminfo.tickerid, '1', ta.ema(close, 20))
```

```
ema5 = request.security(syminfo.tickerid, '5', ta.ema(close, 20))
```

```
ema15 = request.security(syminfo.tickerid, '15', ta.ema(close, 20))
```

```
// Air Support Calculations
```

```
// Slope Calculation: Using atan to find the angle based on price change per bar
```

```
slope15 = math.atan(ta.change(ema15) / syminfo.mintick) * 180 / math.pi
```

```
ema_dist = math.abs(ema1 - ema15)
```

```
ema_bull = ema1 > ema5 and ema5 > ema15 and close > ema1 and slope15 > slope_thresh  
and ema_dist > dist_thresh
```

```
ema_bear = ema1 < ema5 and ema5 < ema15 and close < ema1 and slope15 < -  
slope_thresh and ema_dist > dist_thresh
```

```
plot(show_emas ? ema1 : na, '1m EMA', color = color.new(#5b9cf6, 50))
```

```
plot(show_emas ? ema5 : na, '5m EMA', color = color.new(#2962ff, 0), linewidth = 2)
```

```
plot(show_emas ? ema15 : na, '15m EMA', color = color.new(#1848cc, 0), linewidth = 2)
```

```
// Session HL Tracking
```

```
f_session_hl(sess) =>
```

```
in_sess = not na(time(timeframe.period, sess, "America/New_York"))
```

```
var float h = na
```

```
var float l = na
```

```
if in_sess
```

```
h := na(h) or not in_sess[1] ? high : math.max(high, h)
```

```
l := na(l) or not in_sess[1] ? low : math.min(low, l)
```

```
[h, l]
```

```
// Aligned Colors: Asia Blue, London Yellow
```

```
[asiaH, asiaL] = f_session_hl("1900-0000")
```

```
[lonH, lonL] = f_session_hl("0000-0600")
```

```
[orH, orL] = f_session_hl("0930-0945")
```

```
// --- Liquidity Logic ---
```

```
var float lastHighLiq = na
```

```
var float lastLowLiq = na
```

```
if high >= pdh or high >= pwh or high >= asiaH or high >= lonH or high >= orH
```

```
lastHighLiq := high >= pdh ? pdh : high >= pwh ? pwh : high >= asiaH ? asiaH : high >= lonH  
? lonH : orH
```

```
if low <= pdl or low <= pwl or low <= asiaL or low <= lonL or low <= orL
```

```
lastLowLiq := low <= pdl ? pdl : low <= pwl ? pwl : low <= asiaL ? asiaL : low <= lonL ? lonL  
: orL
```

```
touched_low = ta.barssince(low <= pdl or low <= pwl or low <= asiaL or low <= lonL or low <= orL) <= touch_lookback
```

```
touched_high = ta.barssince(high >= pdh or high >= pwh or high >= asiaH or high >= lonH or high >= orH) <= touch_lookback
```

```
// --- HTF FVG Context ---
```

```
f_get_htf_context() =>
```

```
var float f_t = na, var float f_b = na, var int is_b = 0, var int st = 0
```

```
bull = low > high[2], bear = high < low[2]
```

```
if bull
```

```
    f_t := low, f_b := high[2], is_b := 1, st := 1
```

```
else if bear
```

```
    f_t := low[2], f_b := high, is_b := -1, st := 1
```

```
if st == 1 and ((is_b == 1 and low <= f_t) or (is_b == -1 and high >= f_b))
```

```
    st := 2
```

```
if st == 2 and ((is_b == 1 and close > f_t) or (is_b == -1 and close < f_b))
```

```
    st := 3
```

```
active = (close <= f_t and close >= f_b) or st == 3
```

```
[active, is_b]
```

```
[ctx15, dir15] = request.security(syminfo.tickerid, '15', f_get_htf_context())
```

```
[ctx1h, dir1h] = request.security(syminfo.tickerid, '60', f_get_htf_context())
```

```
[ctx4h, dir4h] = request.security(syminfo.tickerid, '240', f_get_htf_context())
```

```
htf_bull = (use_15m and ctx15 and dir15 == 1) or (use_1h and ctx1h and dir1h == 1) or  
(use_4h and ctx4h and dir4h == 1)
```

```
htf_bear = (use_15m and ctx15 and dir15 == -1) or (use_1h and ctx1h and dir1h == -1) or  
(use_4h and ctx4h and dir4h == -1)
```

```
// --- Signal & Visual Execution ---
```

```
var float[] f_top = array.new_float()
```

```
var float[] f_bot = array.new_float()
```

```
var bool[] f_bul = array.new_bool()
```

```
var int[] f_sta = array.new_int()
```

```
var int[] f_idx = array.new_int()
```

```
var trade_boxes_tp = array.new_box()
```

```
var trade_boxes_sl = array.new_box()
```

```
var trade_lines_ep = array.new_line()
```

```
var trade_lines_sl = array.new_line()
```

```
var trade_lines_tp = array.new_line()
```

```
var trade_times = array.new_int()
```

```
var trade_is_bull = array.new_bool()
```

```
var trade_eps = array.new_float()
```

```
var trade_risks = array.new_float()
```

```
var trade_be_hit = array.new_bool()
```

```
global_alert_triggered := false
```

```
if timeframe.isintraday and timeframe.multiplier == 5
```

```
    if low > high[2]
```

```
        array.push(f_top, low), array.push(f_bot, high[2]), array.push(f_bul, true),  
        array.push(f_sta, 0), array.push(f_idx, bar_index)
```

```

if high < low[2]

    array.push(f_top, low[2]), array.push(f_bot, high), array.push(f_bul, false),
array.push(f_sta, 0), array.push(f_idx, bar_index)

int i = array.size(f_top) - 1

while i >= 0

    float ft = array.get(f_top, i), fb = array.get(f_bot, i), fl = array.get(f_bul, i), fs =
array.get(f_sta, i), fc = array.get(f_idx, i)

    if (fl and low < fb) or (not fl and high > ft)

        array.remove(f_top, i), array.remove(f_bot, i), array.remove(f_bul, i),
array.remove(f_idx, i), array.remove(f_sta, i)

        i := i - 1

        continue

if fs == 0 and bar_index > fc and ((fl and low <= ft) or (not fl and high >= fb))

    array.set(f_sta, i, 1)

if fs == 1 and bar_index > fc

    bool breakout = fl ? close > ft : close < fb

    bool bias    = fl ? htf_bull : htf_bear

    bool liq     = fl ? touched_low : touched_high

    bool emas    = use_ema_filter ? (fl ? ema_bull : ema_bear) : true

if breakout and bias and liq and emas

    float ep = close

    float sl = na

    // SL Calculation based on Mode

    if sl_mode == 'FVG Edge'

```

```

    sl := fl ? fb : ft

else if sl_mode == 'FVG Midpoint'

    sl := (ft + fb) / 2

else // Signal Candle

    sl := fl ? low : high


float risk = math.abs(ep - sl)

float tp = fl ? ep + (risk * rr_mult) : ep - (risk * rr_mult)


// Liquidity Level Targeting with Min RR protection
if fl

    if pdh > ep and (pdh - ep) > (risk * min_rr)

        tp := math.min(tp, pdh)

    if pwh > ep and (pwh - ep) > (risk * min_rr)

        tp := math.min(tp, pwh)

else

    if pdl < ep and (ep - pdl) > (risk * min_rr)

        tp := math.max(tp, pdl)

    if pwl < ep and (ep - pwl) > (risk * min_rr)

        tp := math.max(tp, pwl)


bool is_flow = fl ? (close < lastLowLiq) : (close > lastHighLiq)

bool is_sweep = fl ? (close > lastLowLiq) : (close < lastHighLiq)


label.new(bar_index, fl ? low : high, fl ? "L" : "S", style = fl ? label.style_label_up :
label.style_label_down, color = fl ? COLOR_BULL : COLOR_BEAR, textcolor = color.white,
size = size.small)

```



if is\_flow or is\_sweep

label.new(bar\_index, fl ? low : high, fl ? "▲" : "▼", style = label.style\_label\_center, color = #00000000, textcolor = COLOR\_FLOW, size = size.large, yloc = fl ? yloc.belowbar : yloc.abovebar)

array.push(trade\_boxes\_tp, box.new(bar\_index, fl ? tp : ep, bar\_index + line\_len, fl ? ep : tp, bgcolor = color.new(COLOR\_BULL, 85), border\_color = color.new(COLOR\_BULL, 100)))

array.push(trade\_boxes\_sl, box.new(bar\_index, fl ? ep : sl, bar\_index + line\_len, fl ? sl : ep, bgcolor = color.new(COLOR\_BEAR, 85), border\_color = color.new(COLOR\_BEAR, 100)))

array.push(trade\_lines\_ep, line.new(bar\_index, ep, bar\_index + line\_len, ep, color = COLOR\_ENTRY, style = line.style\_dotted, width = 2))

array.push(trade\_lines\_sl, line.new(bar\_index, sl, bar\_index + line\_len, sl, color = COLOR\_BEAR, style = line.style\_dotted, width = 2))

array.push(trade\_lines\_tp, line.new(bar\_index, tp, bar\_index + line\_len, tp, color = COLOR\_BULL, style = line.style\_dotted, width = 2))

array.push(trade\_times, time)

array.push(trade\_is\_bull, fl)

array.push(trade\_eps, ep)

array.push(trade\_risks, risk)

array.push(trade\_be\_hit, false)

global\_alert\_triggered := true

alert("Footprints - Flow/Sweep ready on " + syminfo.ticker, alert.freq\_once\_per\_bar)

array.remove(f\_top, i), array.remove(f\_bot, i), array.remove(f\_bul, i),  
array.remove(f\_idx, i), array.remove(f\_sta, i)

i := i - 1

```

int k = array.size(trade_times) - 1

while k >= 0

    if (time - array.get(trade_times, k)) > (lookback_hrs * 3600000)

        box.delete(array.remove(trade_boxes_tp, k)),
box.delete(array.remove(trade_boxes_sl, k))

        line.delete(array.remove(trade_lines_ep, k)), line.delete(array.remove(trade_lines_sl,
k)), line.delete(array.remove(trade_lines_tp, k))

        array.remove(trade_times, k), array.remove(trade_is_bull, k), array.remove(trade_eps,
k), array.remove(trade_risks, k), array.remove(trade_be_hit, k)

        k := k - 1

        continue


// Break-Even Logic

if use_be and not array.get(trade_be_hit, k)

    float ep = array.get(trade_eps, k)

    float risk = array.get(trade_risks, k)

    bool is_bull = array.get(trade_is_bull, k)

    float trigger = is_bull ? ep + risk * be_trigger_rr : ep - risk * be_trigger_rr


if (is_bull and high >= trigger) or (not is_bull and low <= trigger)

    line.set_y1(array.get(trade_lines_sl, k), ep)

    line.set_y2(array.get(trade_lines_sl, k), ep)

    box.set_top(array.get(trade_boxes_sl, k), ep)

    box.set_bottom(array.get(trade_boxes_sl, k), ep)

    array.set(trade_be_hit, k, true)

k := k - 1

```

```
// --- Visuals ---
```

```
plot(show_levels ? asiaH : na, "Asia H", color.new(#00bcd4, 50), style = plot.style_linebr)
```

```
plot(show_levels ? asiaL : na, "Asia L", color.new(#00bcd4, 50), style = plot.style_linebr)
```

```
plot(show_levels ? lonH : na, "London H", color.new(#ffeb3b, 50), style = plot.style_linebr)
```

```
plot(show_levels ? lonL : na, "London L", color.new(#ffeb3b, 50), style = plot.style_linebr)
```

```
plot(show_levels ? orH : na, "OR High", color.new(color.purple, 50), style =  
plot.style_linebr)
```

```
plot(show_levels ? orL : na, "OR Low", color.new(color.purple, 50), style = plot.style_linebr)
```