

// 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('Momentum Flow Build - Intention Respect', overlay = true, max_labels_count = 5)

// =====

// ⚠ Sessions & Levels Configuration

// =====

session1_enabled = input.bool(true, title = 'Enable Asia Session', group = '⚠ Asia Session')

session1_time = input.session('1800-0000', title = 'Asia Session Time', group = '⚠ Asia Session')

session2_enabled = input.bool(true, title = 'Enable London Session', group = '⚠ London Session')

session2_time = input.session('0000-0600', title = 'London Session Time', group = '⚠ London Session')

session3_enabled = input.bool(true, title = 'Enable 5m Opening Range', group = '⚠ 5m OR')

session3_time = input.session('0930-0935', title = '5m OR Time', group = '⚠ 5m OR')

session4_enabled = input.bool(true, title = 'Enable 15m Opening Range', group = '⚠ 15m OR')

session4_time = input.session('0930-0945', title = '15m OR Time', group = '⚠ 15m OR')

grpPrev = '● HTF Levels (Daily/Weekly)'

```
prev_enabled = input.bool(true, 'Enable HTF Levels', group = grpPrev)
```

```
grpStyle = '🎨 Display Settings'
```

```
text_size = input.string('Large', 'Narration Text Size', options = ['Small', 'Normal', 'Large',  
'Huge'], group = grpStyle)
```

```
// --- Helper Functions ---
```

```
is_new_bar(sess) =>
```

```
  t = time('D', sess, 'America/New_York')
```

```
  na(t[1]) and not na(t) or t[1] < t
```

```
is_in_session(sess) =>
```

```
  not na(time('D', sess, 'America/New_York'))
```

```
get_session_bounds(session_enabled, session_time) =>
```

```
  var float s_high = na
```

```
  var float s_low = na
```

```
  bool in_sess = is_in_session(session_time)
```

```
  if session_enabled
```

```
    new_bar = is_new_bar(session_time)
```

```
    if in_sess
```

```
      s_low := new_bar ? low : na(s_low) ? low : math.min(s_low, low)
```

```
      s_high := new_bar ? high : na(s_high) ? high : math.max(s_high, high)
```

```
  [s_high, s_low]
```

```
f_size(_s) => _s == 'Small' ? size.small : _s == 'Normal' ? size.normal : _s == 'Large' ?  
size.large : size.huge
```

```
// --- Levels Calculation ---
```

```
[asiaH, asiaL] = get_session_bounds(session1_enabled, session1_time)
```

```
[lonH, lonL] = get_session_bounds(session2_enabled, session2_time)
```

```
[or5H, or5L] = get_session_bounds(session3_enabled, session3_time)
```

```
[or15H, or15L] = get_session_bounds(session4_enabled, session4_time)
```

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

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

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

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

```
is_4pm = hour(time, 'America/New_York') == 16 and minute(time, 'America/New_York') == 0
```

```
var float pdh = na, var float pdl = na, var float pwh = na, var float pwl = na
```

```
if is_4pm or na(pdh)
```

```
    pdh := _raw_pdh, pdl := _raw_pdl, pwh := _raw_pwh, pwl := _raw_pwl
```

```
// =====
```

```
// 📄 Closure Event Narration & Alerts
```

```
// =====
```

```

var string lastEvent = "Waiting for 5m closure event..."

bool is5m = timeframe.isintraday and timeframe.multiplier == 5

if is5m
    string currentEvent = ""

    // Check for "Closure Events" (Crossovers/Crossunders)
    if prev_enabled
        if ta.crossover(close, pdh)
            currentEvent := "5m Candle Closure Above PDH"
        else if ta.crossunder(close, pdl)
            currentEvent := "5m Candle Closure Below PDL"
        else if ta.crossover(close, pwh)
            currentEvent := "5m Candle Closure Above PWH"
        else if ta.crossunder(close, pwl)
            currentEvent := "5m Candle Closure Below PWL"

    if currentEvent == "" and session1_enabled and not is_in_session(session1_time) and not
na(asiaH)
        if ta.crossover(close, asiaH)
            currentEvent := "5m Candle Closure Above Asia High"
        else if ta.crossunder(close, asiaL)
            currentEvent := "5m Candle Closure Below Asia Low"

    if currentEvent == "" and session2_enabled and not is_in_session(session2_time) and not
na(lonH)
        if ta.crossover(close, lonH)

```

```

    currentEvent := "5m Candle Closure Above London High"
else if ta.crossunder(close, lonL)
    currentEvent := "5m Candle Closure Below London Low"

if currentEvent == "" and session3_enabled and not is_in_session(session3_time) and not
na(or5H)
    if ta.crossover(close, or5H)
        currentEvent := "5m Candle Closure Above 5m OR High"
    else if ta.crossunder(close, or5L)
        currentEvent := "5m Candle Closure Below 5m OR Low"

if currentEvent == "" and session4_enabled and not is_in_session(session4_time) and not
na(or15H)
    if ta.crossover(close, or15H)
        currentEvent := "5m Candle Closure Above 15m OR High"
    else if ta.crossunder(close, or15L)
        currentEvent := "5m Candle Closure Below 15m OR Low"

// Process new event
if currentEvent != ""
    lastEvent := currentEvent
    // Trigger alert only on confirmed candle close
    if barstate.isconfirmed
        alert(currentEvent, alert.freq_once_per_bar_close)

// Display Narration
var label eventLabel = na

```

```
if is5m
```

```
    if na(eventLabel)
```

```
        eventLabel := label.new(bar_index, close, lastEvent, xloc = xloc.bar_index, style =  
label.style_label_left, textcolor = #5b9cf6, color = #00000000, textalign = text.align_left,  
size = f_size(text_size))
```

```
    else
```

```
        label.set_xy(eventLabel, bar_index + 2, close)
```

```
        label.set_text(eventLabel, lastEvent)
```

```
        label.set_size(eventLabel, f_size(text_size))
```

```
else
```

```
    if barstate.islast
```

```
        runtime.error("This indicator is designed for the 5-minute timeframe only.")
```