

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

// © LuxAlgo

//@version=6

indicator('MFB - Maximization Prophet - RMF', overlay = true, max\_boxes\_count = 500,  
max\_lines\_count = 500)

// =====

// 🌊 Risk & Setup Settings

// =====

grp\_rmf = ' 🌊 R.M.F. - Risk Management Fielding'

rr\_target = input.float(2.0, 'Profit Target (R)', minval = 1.0, step = 0.1, tooltip = 'The R-multiple  
for the Take Profit.')

rmf\_trigger = input.float(1.5, 'RMF Trail Trigger (R) (Risk Management Fielding)', minval = 1.0,  
step = 0.1, tooltip = 'At this R-multiple, the SL moves to the nearest structural FVG and  
begins trailing.')

sl\_buffer = input.float(0.0, 'SL Buffer (Points)', minval = 0, tooltip = 'Additional points added  
to the FVG boundary for the SL.')

grp\_vis = ' 🎨 Visuals'

show\_zones = input.bool(true, 'Show Key Level Zones', group = grp\_vis)

show\_trade\_zone = input.bool(true, 'Highlight Active Trade Zone', group = grp\_vis)

ext\_right = input.int(15, 'Extend Visuals Right (Bars)', minval = 1, group = grp\_vis)

// =====

// 🇮🇹 Key Level Detection Engine

// =====

```
is_in_sess(sess) => not na(time(timeframe.period, sess, 'America/New_York'))
```

```
is_new_sess(sess) =>
```

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

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

```
var float asiaH = na
```

```
var float asiaL = na
```

```
if is_in_sess('1800-0000')
```

```
  asiaH := is_new_sess('1800-0000') ? high : math.max(nz(asiaH), high)
```

```
  asiaL := is_new_sess('1800-0000') ? low : math.min(nz(asiaL, 1e10), low)
```

```
var float lonH = na
```

```
var float lonL = na
```

```
if is_in_sess('0000-0600')
```

```
  lonH := is_new_sess('0000-0600') ? high : math.max(nz(lonH), high)
```

```
  lonL := is_new_sess('0000-0600') ? low : math.min(nz(lonL, 1e10), low)
```

```
var float or15H = na
```

```
var float or15L = na
```

```
if is_in_sess('0930-0945')
```

```
  or15H := is_new_sess('0930-0945') ? high : math.max(nz(or15H), high)
```

```
  or15L := is_new_sess('0930-0945') ? low : math.min(nz(or15L, 1e10), low)
```

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

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

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

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

```
plot(show_zones ? or15H : na, "15m OR High", color.new(color.yellow, 50), 1,  
plot.style_linebr)
```

```
plot(show_zones ? or15L : na, "15m OR Low", color.new(color.yellow, 50), 1,  
plot.style_linebr)
```

```
// FVG-123 Logic
```

```
fvg_bull = low > high[2]
```

```
fvg_bear = high < low[2]
```

```
fvg_bull_val = high[2]
```

```
fvg_bear_val = low[2]
```

```
// Sweep Logic (Pivots)
```

```
ph = ta.pivohigh(high, 3, 3)
```

```
pl = ta.pivotlow(low, 3, 3)
```

```
var float last_ph = na
```

```
var float last_pl = na
```

```
if not na(ph)
```

```
    last_ph := ph
```

```
if not na(pl)
```

```
    last_pl := pl
```

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

```
// 🏹 Advanced Entry Logic (Flow vs Sweep)
```

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

```
co_pdh = ta.crossover(close, pdh)
co_pwh = ta.crossover(close, pwh)
co_or15h = ta.crossover(close, or15H)
co_lonh = ta.crossover(close, lonH)
co_asiah = ta.crossover(close, asiaH)
co_ph = not na(last_ph) and ta.crossover(close, last_ph)
```

```
cu_pdl = ta.crossunder(close, pdl)
cu_pwl = ta.crossunder(close, pwl)
cu_or15l = ta.crossunder(close, or15L)
cu_lonl = ta.crossunder(close, lonL)
cu_asial = ta.crossunder(close, asiaL)
cu_pl = not na(last_pl) and ta.crossunder(close, last_pl)
```

```
var string long_lvl = ""
var string long_t = ""
if co_pdh
    long_lvl := "PDH", long_t := "Flow"
else
    if co_pwh
        long_lvl := "PWH", long_t := "Flow"
    else
        if co_or15h
            long_lvl := "15m OR High", long_t := "Flow"
        else
            if co_lonh
```

```

    long_lvl := "London High", long_t := "Flow"
else
    if co_asiah
        long_lvl := "Asia High", long_t := "Flow"
    else
        if co_ph
            if low < pdl or low[1] < pdl
                long_lvl := "PDL", long_t := "Sweep"
            else
                if low < pwl or low[1] < pwl
                    long_lvl := "PWL", long_t := "Sweep"
                else
                    if low < or15L or low[1] < or15L
                        long_lvl := "15m OR Low", long_t := "Sweep"
                    else
                        if low < lonL or low[1] < lonL
                            long_lvl := "London Low", long_t := "Sweep"
                        else
                            if low < asiaL or low[1] < asiaL
                                long_lvl := "Asia Low", long_t := "Sweep"
                            else
                                long_lvl := "Local", long_t := "Sweep"
                    else
                        long_lvl := "", long_t := ""

var string short_lvl = ""

```

```

var string short_t = ""

if cu_pdl
    short_lvl := "PDL", short_t := "Flow"
else
    if cu_pwl
        short_lvl := "PWL", short_t := "Flow"
    else
        if cu_or15l
            short_lvl := "15m OR Low", short_t := "Flow"
        else
            if cu_lonl
                short_lvl := "London Low", short_t := "Flow"
            else
                if cu_asial
                    short_lvl := "Asia Low", short_t := "Flow"
                else
                    if cu_pl
                        if high > pdh or high[1] > pdh
                            short_lvl := "PDH", short_t := "Sweep"
                        else
                            if high > pwh or high[1] > pwh
                                short_lvl := "PWH", short_t := "Sweep"
                            else
                                if high > or15H or high[1] > or15H
                                    short_lvl := "15m OR High", short_t := "Sweep"
                                else

```

```

        if high > lonH or high[1] > lonH
            short_lvl := "London High", short_t := "Sweep"
        else
            if high > asiaH or high[1] > asiaH
                short_lvl := "Asia High", short_t := "Sweep"
            else
                short_lvl := "Local", short_t := "Sweep"
        else
            short_lvl := "", short_t := ""

```

```

long_condition = long_t != "" and fvg_bull and close > high[2]
short_condition = short_t != "" and fvg_bear and close < low[2]

```

```

// =====

```

```

// 🟡 R.M.F. State Machine

```

```

// =====

```

```

var float sl_line = na
var float tp_line = na
var float active_entry = na
var int entry_bar = na
var float init_risk = na
var bool is_long = false
var bool at_rmf_trigger = false
var string trade_why = ""
var float[] fvg_levels = array.new_float()

```

```

if (long_condition or short_condition) and na(sl_line)

    is_long := long_condition

    active_entry := close

    entry_bar := bar_index

    sl_line := is_long ? fvg_bull_val - (sl_buffer * syminfo.pointvalue) : fvg_bear_val +
(sl_buffer * syminfo.pointvalue)

    init_risk := math.abs(active_entry - sl_line)

    tp_line := is_long ? active_entry + (rr_target * init_risk) : active_entry - (rr_target * init_risk)

    at_rmf_trigger := false

    trade_why := (is_long ? long_lvl + " " + long_t : short_lvl + " " + short_t)

    array.clear(fvg_levels)

```

```

if not na(active_entry)

    if is_long and fvg_bull

        array.push(fvg_levels, fvg_bull_val)

    else

        if not is_long and fvg_bear

            array.push(fvg_levels, fvg_bear_val)

```

```

if not na(sl_line)

    curr_profit_r = is_long ? (close - active_entry) / init_risk : (active_entry - close) / init_risk

    if curr_profit_r >= rmf_trigger and not at_rmf_trigger

        target_p = is_long ? active_entry + (rmf_trigger * init_risk) : active_entry - (rmf_trigger *
init_risk)

        float nearest = na

        if array.size(fvg_levels) > 0

            float min_d = 1e10

```



```

for i = 0 to array.size(fvg_levels) - 1
    val = array.get(fvg_levels, i)
    if (is_long ? val <= target_p : val >= target_p)
        dist = math.abs(val - target_p)
        if dist < min_d
            min_d := dist
            nearest := val
if na(nearest) and array.size(fvg_levels) > 0
    nearest := array.get(fvg_levels, array.size(fvg_levels) - 1)
if not na(nearest)
    sl_line := nearest
at_rmf_trigger := true

if at_rmf_trigger
    if is_long and fvg_bull and fvg_bull_val > sl_line
        sl_line := fvg_bull_val
    else
        if not is_long and fvg_bear and fvg_bear_val < sl_line
            sl_line := fvg_bear_val
    if (is_long and close < sl_line) or (not is_long and close > sl_line) or (is_long and low <=
sl_line) or (not is_long and high >= sl_line)
        sl_line := na
    if (is_long and high >= tp_line) or (not is_long and low <= tp_line)
        sl_line := na

if na(sl_line)

```

```

active_entry := na

entry_bar := na

trade_why := ""

// =====

// 📊 Top Right Dashboard

// =====

var table info_box = table.new(position.top_right, 2, 5, bgcolor=color.new(color.black, 40),
border_width=1, border_color=color.new(color.gray, 50))

if barstate.islast

    if not na(sl_line)

        table.cell(info_box, 0, 0, "R.M.F. STATUS", text_color=color.white, text_size=size.normal,
bgcolor=color.new(#5b9cf6, 20))

        table.cell(info_box, 1, 0, "ACTIVE", text_color=color.green, text_size=size.normal)

        table.cell(info_box, 0, 1, "Trade Why", text_color=color.gray, text_size=size.normal)

        table.cell(info_box, 1, 1, trade_why, text_color=color.white, text_size=size.normal)

        table.cell(info_box, 0, 2, "SL Anchor", text_color=color.gray, text_size=size.normal)

        table.cell(info_box, 1, 2, at_rmf_trigger ? "Fielded FVG" : "Initial FVG",
text_color=color.yellow, text_size=size.normal)

        table.cell(info_box, 0, 3, "Trail Logic", text_color=color.gray, text_size=size.normal)

        table.cell(info_box, 1, 3, at_rmf_trigger ? "Trailing FVGs" : "Wait 1.5R",
text_color=color.white, text_size=size.normal)

        target_15r = is_long ? active_entry + (rmf_trigger * init_risk) : active_entry - (rmf_trigger *
init_risk)

        dist_pts = math.abs(target_15r - close)

        table.cell(info_box, 0, 4, "To 1.5R Field", text_color=color.gray, text_size=size.normal)

        table.cell(info_box, 1, 4, at_rmf_trigger ? "FIELDDED ✅" : str.format("{0,number,#.##}
pts", dist_pts), text_color=at_rmf_trigger ? color.green : color.white, text_size=size.normal)

```

```

else

    table.cell(info_box, 0, 0, "R.M.F. STATUS", text_color=color.white, text_size=size.normal,
bgcolor=color.new(color.gray, 50))

    table.cell(info_box, 1, 0, "WAITING", text_color=color.gray, text_size=size.normal)

    table.clear(info_box, 0, 1, 1, 4)


// =====

// 🎨 Visuals & Identifiers

// =====

var label entry_diamond = label.new(na, na, "◆", color=#00000000, textcolor=color.yellow,
style=label.style_label_left, size=size.large)

var label entry_txt = label.new(na, na, "", color=#00000000, textcolor=color.yellow,
style=label.style_label_left, size=size.large)

var line sl_ext = line.new(na, na, na, na, color=#f23645, width=1, style=line.style_solid)

var line tp_ext = line.new(na, na, na, na, color=#089981, width=2, style=line.style_dashed)

var label sl_lbl = label.new(na, na, "", color=#f23645, textcolor=color.white,
style=label.style_label_left, size=size.small)

var label tp_lbl = label.new(na, na, "", color=#089981, textcolor=color.white,
style=label.style_label_left, size=size.small)

var box trade_zone = box.new(na, na, na, na, bgcolor=color.new(color.gray, 90),
border_style=line.style_dotted, border_color=color.new(color.gray, 70))

if barstate.islast and not na(sl_line)

    label.set_xy(entry_diamond, bar_index + 1, active_entry)

    label.set_xy(entry_txt, bar_index + 6, active_entry)

    label.set_text(entry_txt, "Entry: " + str.format('{0,number,###}', active_entry))

    line.set_xy1(sl_ext, bar_index, sl_line)

    line.set_xy2(sl_ext, bar_index + ext_right, sl_line)

```

```

label.set_xy(sl_lbl, bar_index + ext_right, sl_line)

label.set_text(sl_lbl, " RMF SL ")

line.set_xy1(tp_ext, bar_index, tp_line)

line.set_xy2(tp_ext, bar_index + ext_right, tp_line)

label.set_xy(tp_lbl, bar_index + ext_right, tp_line)

label.set_text(tp_lbl, " 2R Target (" + str.format('{0,number,###}', tp_line) + ") ")

if show_trade_zone

    box.set_lefttop(trade_zone, entry_bar, active_entry)

    box.set_rightbottom(trade_zone, bar_index, sl_line)

    box.set_bgcolor(trade_zone, is_long ? color.new(#089981, 93) : color.new(#f23645, 93))

else

    if barstate.islast

        label.set_xy(entry_diamond, na, na)

        label.set_xy(entry_txt, na, na)

        line.set_xy1(sl_ext, na, na)

        line.set_xy1(tp_ext, na, na)

        label.set_xy(sl_lbl, na, na)

        label.set_xy(tp_lbl, na, na)

        box.set_lefttop(trade_zone, na, na)

    plot(active_entry, "Active Entry", color.new(color.yellow, 80), 1, plot.style_linebr)

    plotshape(long_condition and na(sl_line[1]), "Long Entry", shape.triangleup,
    location.belowbar, color.green, 0, size = size.small)

    plotshape(short_condition and na(sl_line[1]), "Short Entry", shape.triangledown,
    location.abovebar, color.red, 0, size = size.small)

```