

```

//@version=6

indicator("MFB - 9 EMA retrace", overlay = true, max_lines_count = 500)

// --- Inputs ---

emaFastLen = input.int(9, "Fast EMA")
emaSlowLen = input.int(20, "Slow EMA")
rrRatio = input.float(2.0, "Risk/Reward Ratio", step = 0.5)
lineLength = input.int(25, "Trade Line Length (Bars)", minval = 1, tooltip = "How many bars
into the future the entry/stop/target lines should project")
requireCross = input.bool(true, "Require EMA Cross Before Retrace", tooltip = "If disabled,
signals can trigger on any pullback to the 9 EMA as long as the 9 EMA is above/below the 20
EMA (Continuation trades).")

// --- Core Logic ---

emaFast = ta.ema(close, emaFastLen)
emaSlow = ta.ema(close, emaSlowLen)

// State variables to track the setup phases
// 0 = Neutral, 1 = Armed (Crossed or Trending), 2 = Retraced
var int longState = 0
var int shortState = 0

// Arm states on crossovers or continuous trend (if cross is not required)
if ta.crossover(emaFast, emaSlow)
    longState := 1
    shortState := 0
else if ta.crossunder(emaFast, emaSlow)

```

```
shortState := 1
longState := 0
else if not requireCross
  if emaFast > emaSlow and longState == 0
    longState := 1
  if emaFast < emaSlow and shortState == 0
    shortState := 1

// Check for retracements (price touching/crossing the fast EMA after a cross)
if longState == 1 and low <= emaFast
  longState := 2
if shortState == 1 and high >= emaFast
  shortState := 2

// Check for closure beyond the fast EMA to trigger the signal
longSignal = longState == 2 and close > emaFast
shortSignal = shortState == 2 and close < emaFast

// Reset state after signal fires to avoid consecutive spam signals
if longSignal
  longState := 0
if shortSignal
  shortState := 0

// --- Visual Elements ---
// Plot EMAs
```

```

plot(emaFast, "Fast EMA", color = color.blue)
plot(emaSlow, "Slow EMA", color = color.red)

// Plot Signals

plotshape(longSignal, "Long Entry", shape.triangleup, location.belowbar,
color.new(color.green, 0), size = size.small)

plotshape(shortSignal, "Short Entry", shape.triangledown, location.abovebar,
color.new(color.red, 0), size = size.small)

// Draw Risk/Reward visualization on signals

if longSignal

    entryPrice = close

    stopLoss = emaSlow

    risk = entryPrice - stopLoss

    takeProfit = entryPrice + (risk * rrRatio)

    if risk > 0

        line.new(bar_index, entryPrice, bar_index + lineLength, entryPrice, color = color.gray,
style = line.style_dotted, width = 2)

        line.new(bar_index, stopLoss, bar_index + lineLength, stopLoss, color = color.red, style
= line.style_dotted, width = 2)

        line.new(bar_index, takeProfit, bar_index + lineLength, takeProfit, color = color.green,
style = line.style_dotted, width = 2)

if shortSignal

    entryPrice = close

    stopLoss = emaSlow

```

```
risk = stopLoss - entryPrice

takeProfit = entryPrice - (risk * rrRatio)

if risk > 0

    line.new(bar_index, entryPrice, bar_index + lineLength, entryPrice, color = color.gray,
style = line.style_dotted, width = 2)

    line.new(bar_index, stopLoss, bar_index + lineLength, stopLoss, color = color.red, style
= line.style_dotted, width = 2)

    line.new(bar_index, takeProfit, bar_index + lineLength, takeProfit, color = color.green,
style = line.style_dotted, width = 2)

// --- Alerts ---
alertcondition(longSignal or shortSignal, "9 EMA retrace Alert", "9 EMA retrace")
```