

```

// This source code is subject to the terms of the Mozilla Public License 2.0 at
https://mozilla.org/MPL/2.0/
// © OK

//@version=5
strategy("FTMO Rules Monitor", overlay=true, initial_capital=10000,
default_qty_type=strategy.percent_of_equity, default_qty_value=1,
calc_on_every_tick=true, process_orders_on_close=true)

// Account Settings
var string G1 = "FTMO Account Settings"
accountSize = input.float(10000, "Account Size ($)", group=G1, options=[10000,
25000, 50000, 100000, 200000], tooltip = "Set the account size in the Properties
tab with the same value")
isChallengePhase = input.bool(true, "Is Challenge Phase?", group=G1)

// Define account-specific parameters
getMaxDailyLoss(size) =>
    switch size
        10000 => -500
        25000 => -1250
        50000 => -2500
        100000 => -5000
        => -10000 // 200000

getMaxTotalLoss(size) =>
    switch size
        10000 => -1000
        25000 => -2500
        50000 => -5000
        100000 => -10000
        => -20000 // 200000

getProfitTarget(size) =>
    switch size
        10000 => 1000
        25000 => 2500
        50000 => 5000
        100000 => 10000
        => 20000 // 200000

// Strategy Parameters
var string G2 = "Strategy Parameters"
volumeThreshold = input.float(1.5, "Volume Threshold", group=G2, tooltip = "Set
the volume threshold in the Properties tab with the same value")
riskPercent = input.float(1.0, "Risk Percent", minval=0.1, maxval=10.0, step=0.1,
group=G2, tooltip = "Set the risk percent in the Properties tab with the same
value")*0.01
atrPeriod = input.int(14, "ATR Period", group=G2, tooltip = "Set the ATR period in
the Properties tab with the same value")

```

```

atrMultiplier = input.float(2.0, "ATR Multiplier", minval=0.5, maxval=5.0,
step=0.1, group=G2, tooltip = "Set the ATR multiplier in the Properties tab with
the same value")

// Supertrend 1 Parameters
var string ST1 = "Supertrend 1 Settings"
st1_atrPeriod = input.int(10, "ATR Period", minval=1, maxval=50, group=ST1)
st1_factor = input.float(2, "Factor", minval=0.5, maxval=10, step=0.5, group=ST1)

// Supertrend 2 Parameters
var string ST2 = "Supertrend 2 Settings"
st2_atrPeriod = input.int(14, "ATR Period", minval=1, maxval=50, group=ST2)
st2_factor = input.float(3, "Factor", minval=0.5, maxval=10, step=0.5, group=ST2)

// Calculate Supertrends
[supertrend1, direction1] = ta.supertrend(st1_factor, st1_atrPeriod)
[supertrend2, direction2] = ta.supertrend(st2_factor, st2_atrPeriod)

// Volume condition
volMA = ta.sma(volume, 20)
isHighVol = volume > volMA * volumeThreshold

// Entry conditions
longCondition = direction1 == -1 and direction2 == -1 and direction1[1] == 1 and
isHighVol
shortCondition = direction1 == 1 and direction2 == 1 and direction1[1] == -1 and
isHighVol

// Position sizing based on ATR
atr = ta.atr(atrPeriod)
stoploss = atrMultiplier * atr
positionSize = riskPercent// * strategy.equity

// Execute trades
if longCondition
    strategy.entry("Long", strategy.long, qty=positionSize)
    strategy.exit("Long Exit", "Long", stop=close - stopLoss, limit=close +
stopLoss*2)

if shortCondition
    strategy.entry("Short", strategy.short, qty=positionSize)
    strategy.exit("Short Exit", "Short", stop=close + stopLoss, limit=close -
stopLoss*2)

// Optional: Plot Supertrends
plot(supertrend1, "Supertrend 1", color = direction1 == -1 ? color.green :
color.red, linewidth=3)
plot(supertrend2, "Supertrend 2", color = direction2 == -1 ? color.lime :
color.maroon, linewidth=3)

// FTMO Rules Tracking

```

```

var float maxDailyLoss = getMaxDailyLoss(accountSize)
var float maxLoss = getMaxTotalLoss(accountSize)
var float profitTarget = getProfitTarget(accountSize)
var int minTradingDays = 4

// Track daily and total performance
var float previousDayEquity = strategy.equity
var int tradingDays = 0
var int consecutiveTradingDays = 0
var int maxConsecutiveDays = 0
var float dailyPnL = 0.0
var float maxDailyDrawdown = 0.0
var float maxDrawdown = 0.0
var int lastTradeDay = 0 // Store the last day we counted

// Reset tracking on new day if no trades were made
if ta.change(dayofmonth)
    if dayofmonth != lastTradeDay + 1 and lastTradeDay > 0
        tradingDays := 0
        consecutiveTradingDays := 0
        previousDayEquity := strategy.equity
        dailyPnL := 0.0

// Update tracking variables when new trades are opened
if strategy.opentrades != strategy.opentrades[1] and dayofmonth != lastTradeDay
    tradingDays := tradingDays + 1

    // Update consecutive days
    if dayofmonth == lastTradeDay + 1 or lastTradeDay == 0
        consecutiveTradingDays := consecutiveTradingDays + 1
    else
        consecutiveTradingDays := 1

    // Update max consecutive days if needed
    if consecutiveTradingDays > maxConsecutiveDays
        maxConsecutiveDays := consecutiveTradingDays

    lastTradeDay := dayofmonth

// Calculate current daily PnL
dailyPnL := strategy.netprofit - strategy.netprofit[1]
if dailyPnL < maxDailyDrawdown
    maxDailyDrawdown := dailyPnL

// Track peak equity
var float peakEquity = strategy.equity
if strategy.equity > peakEquity
    peakEquity := strategy.equity

// Calculate drawdown from peak
curDrawdown = strategy.equity - peakEquity

```

```

if curDrawdown < maxDrawdown
    maxDrawdown := curDrawdown

// Rules Status
bool isDailyLossOK = maxDailyDrawdown > maxDailyLoss
bool isTotalLossOK = maxDrawdown > maxLoss
bool isProfitTargetMet = strategy.netprofit >= profitTarget
bool isMinTradingDaysMet = tradingDays >= minTradingDays

// Close all positions when challenge is completed
if isProfitTargetMet and isMinTradingDaysMet and isDailyLossOK and isTotalLossOK
    strategy.close_all("Challenge Complete")

    alert("🏆 FTMO Challenge Complete!", alert.freq_once_per_bar_close)

    if barstate.islast
        label.new(bar_index, high, "🏆 FTMO Challenge Complete!",
            color=color.green,
            style=label.style_label_down,
            textcolor=color.white,
            size=size.large)

// Color scheme settings
var string G3 = "Visual Settings"
colorScheme = input.string("Auto", "Color Scheme", options=["Auto", "Dark",
"Light"], group=G3)

// Dynamic color function
getColorScheme() =>
    bgColor = chart.bg_color
    auto_isDark = (color.r(bgColor) + color.g(bgColor) + color.b(bgColor))/3 < 128

    isDarkMode = switch colorScheme
        "Auto" => auto_isDark
        "Dark" => true
        "Light" => false

    [isDarkMode ? color.white : color.black, isDarkMode ? color.rgb(66, 66, 66,
50) : color.rgb(240, 240, 240, 50), isDarkMode ? color.rgb(66, 66, 66) :
color.rgb(220, 220, 220)]

[textColor, tableBgColor, headerBgColor] = getColorScheme()

// Create status table
var table statusTable = table.new(position.top_right, 4, 6, bgcolor=tableBgColor)

var string PASSED = "✓"
var string FAILED = "✗"

if barstate.islast
    // Header spanning all columns

```

```

    table.cell(statusTable, 0, 0, "FTMO Rules Monitor", bgcolor=headerBgColor,
text_color=textColor, text_size=size.normal)
    table.merge_cells(statusTable, 0, 0, 3, 0)

    // Column headers
    table.cell(statusTable, 0, 1, "Rule", bgcolor=headerBgColor,
text_color=textColor)
    table.cell(statusTable, 1, 1, "Target", bgcolor=headerBgColor,
text_color=textColor)
    table.cell(statusTable, 2, 1, "Current", bgcolor=headerBgColor,
text_color=textColor)
    table.cell(statusTable, 3, 1, "Status", bgcolor=headerBgColor,
text_color=textColor)

    // Daily Loss Rule
    table.cell(statusTable, 0, 2, "Max Daily Loss", text_color=textColor)
    table.cell(statusTable, 1, 2, str.toString(maxDailyLoss),
text_color=textColor)
    table.cell(statusTable, 2, 2, str.toString(maxDailyDrawdown, "#"),
text_color=textColor)
    table.cell(statusTable, 3, 2, isDailyLossOK ? PASSED : FAILED,
text_color=isDailyLossOK ? color.green : color.red, text_size=size.large)

    // Total Loss Rule
    table.cell(statusTable, 0, 3, "Max Total Loss", text_color=textColor)
    table.cell(statusTable, 1, 3, str.toString(maxLoss), text_color=textColor)
    table.cell(statusTable, 2, 3, str.toString(maxDrawdown, "#"),
text_color=textColor)
    table.cell(statusTable, 3, 3, isTotalLossOK ? PASSED : FAILED,
text_color=isTotalLossOK ? color.green : color.red, text_size=size.large)

    // Profit Target
    table.cell(statusTable, 0, 4, "Profit Target", text_color=textColor)
    table.cell(statusTable, 1, 4, str.toString(profitTarget, "#"),
text_color=textColor)
    table.cell(statusTable, 2, 4, str.toString(strategy.netprofit, "#"),
text_color=textColor)
    table.cell(statusTable, 3, 4, isProfitTargetMet ? PASSED : FAILED,
text_color=isProfitTargetMet ? color.green : color.red, text_size=size.large)

    // Minimum Trading Days
    table.cell(statusTable, 0, 5, "Min Trading Days", text_color=textColor)
    table.cell(statusTable, 1, 5, str.toString(minTradingDays),
text_color=textColor)
    table.cell(statusTable, 2, 5, str.toString(tradingDays, "#"),
text_color=textColor)
    table.cell(statusTable, 3, 5, isMinTradingDaysMet ? PASSED : FAILED,
text_color=isMinTradingDaysMet ? color.green : color.red, text_size=size.large)

    // Add warning label for failed conditions
    if barstate.islast and (not isDailyLossOK or not isTotalLossOK)

```

```
label.new(bar_index, high, "⚠ Risk Limits Exceeded",  
color=color.new(color.red, 0), style=label.style_label_down,  
textcolor=color.white)
```