Levels Feature

Executive Summary

The Problem

Trading strategies need to know **when** and **where** to trade:

Dimension	Current Solution	Gap
When (Time)	Phases	Fully implemented
Where (Price)	Manual conditions	No structured approach

Traders manually code price-level logic into entry/exit conditions, leading to:

- Repetitive, error-prone DSL expressions
- No reusability across strategies
- No visualization of key price zones
- Difficult to test "what if price is near support?"

The Solution: Levels

Levels are a first-class concept for price-based filtering, parallel to Phases.

```
Phases  = Time axis filtering  (WHEN to trade)
Levels  = Price axis filtering (WHERE to trade)
```

Example Usage

```yaml
entrySettings:
  condition: RSI(14) < 30
  phaseSettings:
    requiredPhaseIds: [uptrend]      # Only during uptrends
  levelSettings:
    requiredLevelIds: [fib-618]      # Only at 61.8% retracement
    excludedLevelIds: [near-ath]     # Not near all-time high
```

Level Types (14 Total)

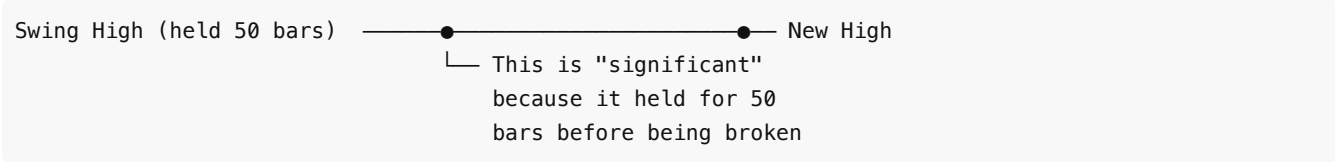
Type	Description	Use Case
ATH/ATL	All-time high/low with dominance tracking	"40% down from ATH"
Fibonacci	Retracements & extensions	Entry at golden ratio pullbacks
HTF Structure	Weekly/monthly/yearly highs/lows/opens	Institutional reference points
S/R Zones	Auto-detected support/resistance	Natural price structure
Round Numbers	Psychological levels ($50k, $100k)	Market psychology barriers

Projections	Calculated price zones	Custom target zones
Ray-Based	Trendline proximity	Dynamic S/R from rays
Custom	User-defined static zones	Manual important levels
FVG	Fair Value Gap zones (ICT)	Imbalance retracement entries
Order Blocks	Institutional entry zones (ICT)	Smart money footprints
Liquidity	Stop-loss clusters	Sweep prediction
Structure Break	BOS price levels	Trend change zones
Condition Projection	Historical condition locations	"Where did X happen?"

Key Capabilities

1. Swing Detection with Dominance

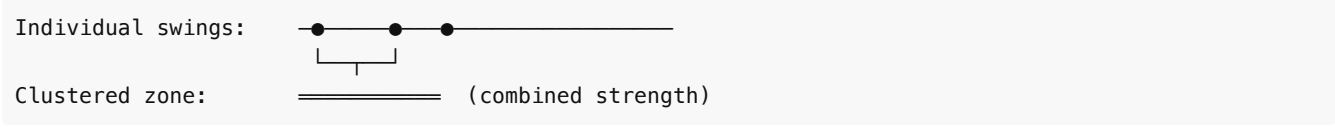
Identifies significant price pivots by measuring how long they held:



Dominance = number of bars a swing held before being exceeded. Higher dominance = more significant level.

2. Automatic Zone Clustering

Groups nearby swings into consolidated S/R zones:



3. Recency-Weighted Strength

Recent swings matter more than old ones:

```
strength = Σ (dominanceBars × recencyWeight)

where recencyWeight = 1.0 / (1.0 + barsAgo / halfLifeBars)
```

4. Multi-Timeframe Support

Every level specifies its timeframe:

```yaml
id: fib-618-daily
type: fib
timeframe: 1d    # Calculated from daily candles

id: sr-zones-4h
```

```
type: sr
timeframe: 4h    # Swings detected on 4h chart
```

Same level type on different timeframes = different zones.

ICT / Smart Money Concepts

Advanced institutional trading patterns integrated as level types:

Fair Value Gaps (FVG)

Price imbalances where gaps tend to get filled:

```
Bullish FVG:
     ┌──┐ candle 3
     │  │
     │  │  ← GAP
     └──┘

     ┌──┐ candle 2
     │██│ (impulse)
     └──┘

     ┌──┐ candle 1
     └──┘
```

Order Blocks

Last opposing candle before institutional move:

```
     ┌──┐ impulse up
     │██│
     │██│
OB → └──┘ ← last DOWN candle
     ┌──┐    before move
     │  │
```

Liquidity Pools

Stop-loss clusters that get swept:

```
═══════════════════════  ← Equal highs = obvious stops
     /\      /\      /\
    /  \    /  \    /  \
```

Strategy Integration

Levels integrate seamlessly with existing strategy structure:

```yaml
# Entry: Only at Fib 61.8%, not near ATH
entrySettings:
  condition: RSI(14) < 30
  levelSettings:
    requiredLevelIds: [fib-618]
    excludedLevelIds: [near-ath]
```

```yaml
# Exit: Take profit at weekly resistance
exitSettings:
  zones:
    - name: Take Profit
      levelSettings:
        requiredLevelIds: [weekly-high]
      exitImmediately: true
```

DSL Extensions

New functions for price structure analysis:

Function	Returns	Example
`ATH(mode, dominance)`	ATH price	`ATH(latest, 50)`
`ATL(mode, dominance)`	ATL price	`ATL(latest, 100)`
`SWING_HIGH(lookback, dominance)`	Recent swing high	`SWING_HIGH(100, 20)`
`SWING_LOW(lookback, dominance)`	Recent swing low	`SWING_LOW(200, 50)`
`FIB(ratio, high, low)`	Fib level price	`FIB(0.618, ATH, ATL)`
`WEEKLY_HIGH/LOW/OPEN`	HTF reference	`WEEKLY_HIGH(1)`
`IN_LEVEL(id)`	Boolean	`IN_LEVEL(fib-618)`

Implementation Phases

Phase	Scope	Deliverables
1	Core Infrastructure	Level model, persistence, zone calculations
2	DSL Extensions	ATH/ATL/FIB/SWING functions in parser
3	Calculation Engine	SwingDetector, ZoneClusterer, LevelCalculator
4	Strategy Integration	BacktestEngine integration, trade analytics
5	UI	Level list, editor, chart overlay
6	MCP/API	CRUD endpoints, evaluation endpoint

Built-in Levels (Presets)

ID	Type	Description
`fib-382`	Fibonacci	38.2% retracement
`fib-500`	Fibonacci	50% retracement
`fib-618`	Fibonacci	61.8% retracement (golden ratio)

`near-ath`	ATH	Within 5% of all-time high
`near-atl`	ATL	Within 5% of all-time low
`previous-ath`	ATH	Previous ATH (broken, now support)
`weekly-high`	HTF	Current week high
`weekly-low`	HTF	Current week low
`monthly-high`	HTF	Current month high
`monthly-low`	HTF	Current month low
`auto-resistance`	S/R	Auto-detected resistance zones
`auto-support`	S/R	Auto-detected support zones

File Structure

```
~/.tradery/
├── levels/
│   ├── {id}/
│   │   └── level.yaml
│   └── ...
├── strategies/...
├── phases/...
```

Built-in levels: `src/main/resources/levels/`

MCP Tools

Tool	Purpose
`tradery_list_levels`	List all levels
`tradery_get_level`	Get level configuration
`tradery_create_level`	Create new level
`tradery_update_level`	Update existing level
`tradery_delete_level`	Delete level
`tradery_eval_level`	Check if level active at current price

Value Proposition

Before	After
Manual price conditions in DSL	Reusable level definitions
No visualization	Chart overlay showing zones

Repetitive code	Single definition, multiple strategies
Hard to test	`tradery_eval_level` for instant feedback
No institutional patterns	ICT/SMC concepts built-in

Catalog: 130+ Level Concepts

The implementation supports a comprehensive catalog of price levels:

Category	Count	Examples
Price Extremes	10	ATH, ATL, swing highs/lows
Fibonacci	6	Retracements, extensions, clusters
Higher Timeframe	8	Weekly/monthly opens, session opens
Volume Profile	12	POC, VAH/VAL, VWAP
Pivot Points	5	Traditional, Fibonacci, Camarilla
Smart Money (ICT)	8	Order blocks, FVG, liquidity pools
Gap Levels	6	Overnight, weekend, CME gaps
Psychological	5	Round numbers, options strikes
Event-Based	7	FOMC range, crash lows
Advanced Orderflow	12	CVD extremes, absorption zones
Astronomical	7	Full moon price, eclipse prices
Projections	8	% from ATH, measured moves

Priority for initial implementation: ATH/ATL, Fibonacci, HTF, S/R zones, Round numbers

---

Architecture Parallel

Aspect	Phases	Levels
Axis	Time	Price
Active when	Condition true	Price in zone
Timeframe	Evaluation resolution	Calculation resolution
Categories	Trend, Session, Calendar	Fib, ATH, HTF, SR, SMC
Integration	`phaseSettings`	`levelSettings`

Document generated from LEVELS_IMPLEMENTATION_PLAN.md