Portfolio Lab Manual

1) What Portfolio Lab is

Portfolio Lab is a multi-asset backtesting workbench built around a clear separation of responsibilities:

- **Strategies** decide *what* to trade and *when* (buy/sell/close signals across one or many tickers).
- **Portfolio sizing / allocation (Sizers)** decide *how much* to trade (position sizing, leverage caps, drawdown scaling, equal-weight, volatility targeting, etc.).
- **Engine settings** define the simulation environment (date range, benchmark, initial cash, commission, slippage).
- **Reporting** produces plots and summary statistics (e.g., equity curve, returns histogram) and run logs (as shown in the UI).

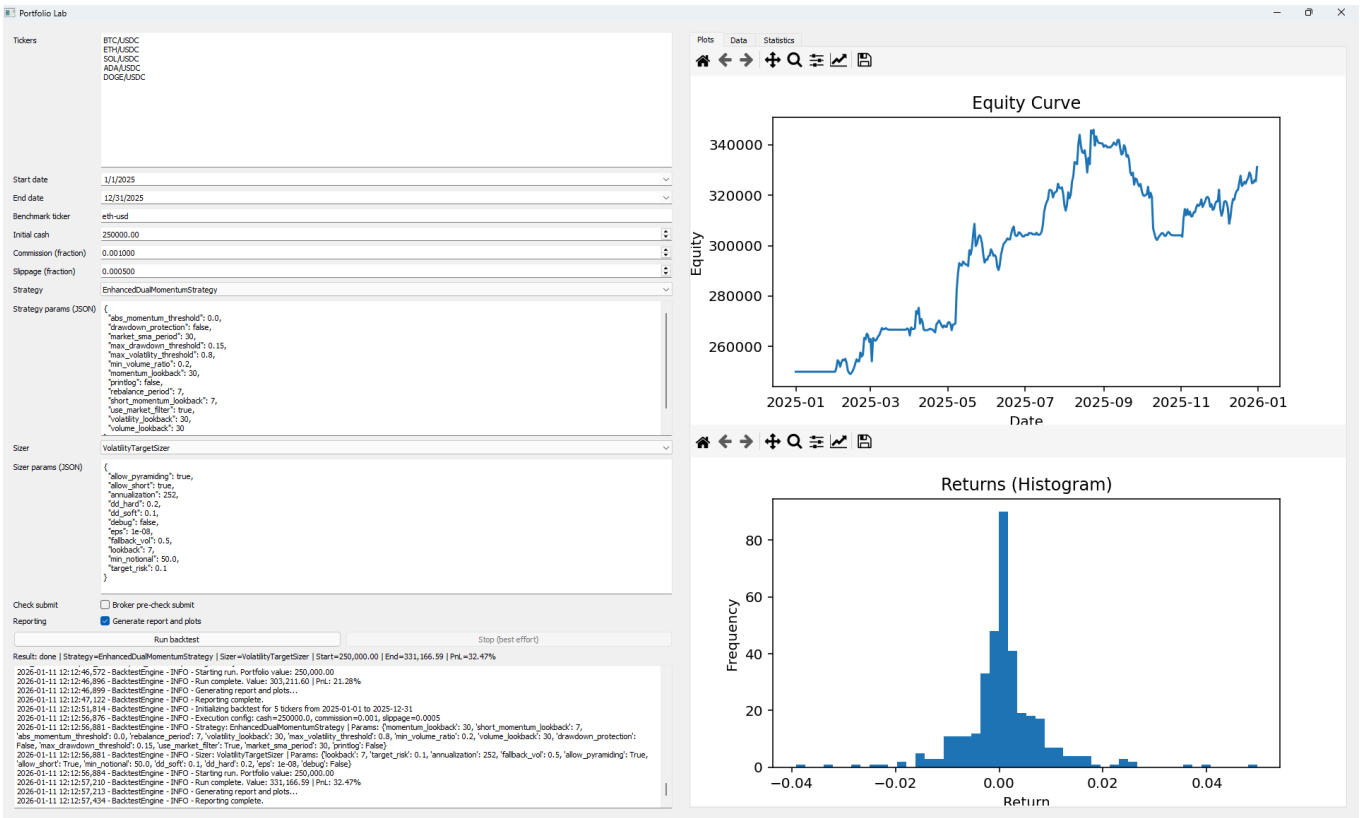
This manual is written to match the components present in your current Strategy and Sizer libraries.

---

2) Quick start workflow (end-to-end)

1. **Enter tickers** (one per line) in the Tickers box.
   Example (from your UI): `BTC/USDC`, `ETH/USDC`, `SOL/USDC`, etc.
2. **Set Start date / End date** for the backtest window.
3. **Set Benchmark ticker** (used for comparison/reporting).
4. **Set Initial cash** (starting portfolio equity).

5. **Set Commission (fraction)** and **Slippage (fraction)**
   - Commission example: `0.001` means 0.10% per fill.
   - Slippage example: `0.0005` means 0.05% execution slippage.
6. **Select a Strategy** from the dropdown.
7. **Adjust Strategy params (JSON)** as needed.
8. **Select a Sizer (allocation method)** from the dropdown.
9. **Adjust Sizer params (JSON)** as needed.
10. (Optional) Enable **Generate report and plots**.
11. Click **Run backtest**.
12. Review results in:
    - **Plots** tab (equity curve, return distribution, etc.)
    - **Data** tab (time series / trades / returns, depending on your build)
    - **Statistics** tab (performance and risk metrics, depending on your build)
    - **Result / logs** panel (execution trace, warnings, final portfolio value)



3) UI field reference (left panel)

Tickers

- Input format is determined by your data adapter. Use the exact symbol format your data loader expects.
- One ticker per line is the recommended convention for multi-asset portfolios.

Dates

- **Start date / End date** define the backtest interval.
- Many indicators require a warm-up period; if the date range is short, you may see fewer trades.

Benchmark ticker

- Used for comparison in reporting and/or for “market filter” logic in some strategies (depending on implementation).

Initial cash

- Starting equity for the backtest (e.g., `250000`).

Commission (fraction) and Slippage (fraction)

- Both act as execution friction and can materially change results.
- If you are validating raw signal quality, start with small but non-zero values to avoid over-optimistic conclusions.

Strategy and Strategy params (JSON)

- Select one of the available strategies (see Section 5).
- Params are editable as a JSON object that maps parameter names to values.
- Operational rule: if you mistype a parameter name, most Backtrader-based stacks will silently ignore it or throw a parsing/config error depending on your glue code. Use only documented parameter keys.

Sizer and Sizer params (JSON)

- Select one of the available sizing/allocation engines (see Section 6).
- Sizers control risk budgeting, leverage clamps, drawdown scaling, and long/short mechanics.

Reporting / Controls

- **Generate report and plots**: enables the plotting and stats pipeline.
- **Run backtest**: executes the engine with the chosen configuration.

- **Stop (best effort)**: attempts to interrupt execution safely.
- **Result / logs**: displays engine messages (start/end portfolio value, strategy chosen, parameter echo, and any runtime notices).

---

4) Core concepts you must get right

A) Strategy vs. Sizer (who decides position size?)

In Portfolio Lab, there are two sizing patterns:

1. **Sizer-driven sizing** (recommended for consistent portfolio research)
   The strategy issues orders without explicit `size=...`; the selected Sizer computes position size.
   Examples that are sizer-driven in your library:
   - `DualEMACrossStrategy` (explicitly notes “no size= here -> sizer decides”)
   - `AdaptiveMAVolatilityStrategy`
   - `ADXTrendStrengthWithVWAPFilter`
   - `BBSqueezeBreakout`
   - `EnhancedVolumeProfileStrategy`
   - `EnhancedDualMomentumStrategy` (orders placed without explicit order size)
2. **Strategy-internal sizing** (Sizer is partially or fully bypassed)
   The strategy computes size and passes `size=...` in orders (risk-per-trade sizing, etc.).
   In your library, this applies to:
   - `AdaptiveMomentumSqueezeStrategy` (fixed-fraction risk sizing using ATR stop distance)
   - `VortexTrendCapturePortfolio` (ATR risk sizing, portfolio-level max positions)

Practical implication:

- If you pick an advanced Sizer (risk parity, volatility targeting, Kelly) but your strategy hardcodes `size=`, your Sizer will not control exposure the way you expect.

B) Long/short order mechanics

Your Sizers explicitly implement mechanically correct handling such as:

- **Buy covers a short** (before opening a new long)
- **Sell closes a long** (before opening a new short)
- Optional rules for pyramiding and max positions
  These mechanics are encoded in the Sizer implementations.

5) Trading strategy catalog (built-in)

This section documents the strategies currently present in your `strategies.py` library.

5.1 AccumulationDistributionStrategy

Concept: Divergence-based trading using an Accumulation/Distribution-style line derived from CLV (close location value) and volume. Detects swing highs/lows and divergence versus price; uses a trend filter and stop loss.

Default parameters

- `swing_period` : 7
- `divergence_lookback` : 30
- `trend_period` : 30
- `stop_loss_pct` : 0.02

When to use

- Mean-reversion / reversal research where divergence confirmation is central.

Sizer pairing

- Works well with conservative sizing (EqualWeight, MaxPositionsEqualWeight) or volatility targeting.

5.2 DualEMACrossStrategy

Concept: Classic fast/slow EMA crossover with bracket orders (stop-loss and take-profit). Designed to run across multiple assets; explicitly leaves sizing to the Sizer (“no size= here -> sizer decides”).

Default parameters

- `fast_period` : 7
- `slow_period` : 30
- `allow_short` : True
- `stop_loss` : 0.05
- `take_profit` : 0.10
- `printlog` : True

Sizer pairing

- If you want portfolio consistency, pair with:
  - Risk parity sizing (`RiskParityVolTargetSizer`)
  - Volatility targeting (`VolatilityTargetSizer`)
  - Equal-weight with max positions (`MaxPositionsEqualWeightSizer`)

5.3 AdaptiveMAVolatilityStrategy

Concept: KAMA-based adaptive moving average with volatility bands; can run in *pullback* or *breakout* mode; includes optional trailing behavior and detailed order logging.

Default parameters

- `kama_period` : 20
- `fast_ema` : 2
- `slow_ema` : 30
- `vol_period` : 20
- `vol_mult` : 2.0
- `trade_mode` : `"pullback"` (`"pullback"` or `"breakout"`)
- `trail_perc` : 0.05
- `printlog` : True

Sizer pairing

- Strong candidate for volatility-based sizing (VolatilityTarget, RiskParityVolTarget).

5.4 AdaptiveMomentumSqueezeStrategy

Concept: Squeeze + momentum cross system using KAMA/EMA cross and Bollinger bandwidth. Uses **internal fixed-fractional risk sizing** based on ATR stop distance and issues `StopTrail` orders. Because it sets explicit `size=` , it largely bypasses external sizers.

Default parameters

- `kama_period` : 14
- `ema_period` : 20
- `bb_period` : 20

- `bb_std` : 2.0
- `squeeze_threshold` : 0.10
- `trail_percent` : 0.05
- `atr_period` : 14
- `risk_per_trade` : 0.02
- `debug` : False

When to use

- When you want each position sized by a consistent “risk at stop” budget without relying on external portfolio allocators.

5.5 ADXTrendStrengthWithVWAPFilter

Concept: Trend regime confirmation using ADX and DI (+DI/-DI), filtered by VWAP positioning, with confirmation bars and trailing stop behavior. Multi-asset capable.

Default parameters

- `adx_period` : 14
- `adx_threshold` : 25
- `boll_period` : 20
- `boll_devfactor` : 2
- `confirmation_bars` : 3
- `trail_percent` : 0.02
- `vwap_period` : 50
- `printlog` : True

Sizer pairing

- Pairs well with volatility-targeted or risk-parity sizing, especially if you run it across correlated assets.

5.6 BBSqueezeBreakout

Concept: Multi-asset Bollinger “squeeze percentile rank” breakout with optional OBV confirmation, designed to trade volatility expansions after compressions.

Default parameters

- `bb_period` : 20
- `bb_devfactor` : 2.0
- `squeeze_lookback` : 40
- `squeeze_threshold_pct` : 10.0
- `obv_lookback` : 10
- `use_volume_confirm` : True
- `printlog` : True

Sizer pairing

- VolatilityTargetSizer is a natural fit, as breakouts often concentrate tail risk.

5.7 VortexTrendCapturePortfolio

Concept: Portfolio version of a Vortex-based trend system. Implements:

- Per-asset Vortex cross signals
- Macro trend filter (long-term MA)
- ATR stability filter (ATR % threshold)
- Portfolio constraint: `max_positions`
- **Internal ATR risk sizing** via `risk_per_trade` and `atr_stop_multiplier`

Default parameters

- `vortex_period` : 30
- `long_term_ma_period` : 30
- `atr_period` : 7
- `atr_threshold` : 0.1
- `atr_stop_multiplier` : 2.0
- `max_positions` : 6
- `risk_per_trade` : 0.05
- `min_size` : 0.1
- `allow_shorts` : True
- `log` : False

Sizer note

- Because the strategy sizes entries with `size=`, external Sizers will not drive sizing in the usual way.

5.8 EnhancedVolumeProfileStrategy

Concept: Volume-profile style regime logic using:

- Rolling price/volume history
- Volume profile bins
- VPOC/VAH/VAL levels
- ATR-based distance thresholds
- Trend MA and volume confirmation
- Trailing stop management

Default parameters

- `profile_period`: 30
- `signal_period`: 7
- `value_area_pct`: 80
- `price_bins`: 30
- `smooth_bins`: 3
- `atr_period`: 14
- `vpoc_atr_mult`: 1.0
- `va_atr_mult`: 0.6
- `decay_factor`: 0.95
- `volume_confirm_mult`: 1.2
- `trend_period`: 30
- `pullback_bars`: 3
- `trail_stop_pct`: 0.02

Sizer pairing

- Start with MaxPositionsEqualWeightSizer to prevent over-concentration, then graduate to VolatilityTargetSizer.

5.9 EnhancedDualMomentumStrategy

Concept: Portfolio dual momentum selection:

- Computes momentum over a long lookback and a short lookback
- Selects one “best long” and one “best short” candidate
- Optional market filter using SMA
- Optional drawdown protection
- Includes volatility and volume gating logic

Default parameters

- `momentum_lookback`: 30
- `short_momentum_lookback`: 7
- `abs_momentum_threshold`: 0.0
- `rebalance_period`: 7
- `volatility_lookback`: 30
- `max_volatility_threshold`: 0.8
- `min_volume_ratio`: 0.2
- `volume_lookback`: 30
- `drawdown_protection`: False
- `max_drawdown_threshold`: 0.15
- `use_market_filter`: True
- `market_sma_period`: 30
- `printlog`: False

Sizer pairing

- Works particularly well with portfolio-aware sizing (risk parity / vol targeting) because the strategy itself focuses on selection and timing rather than notional exposure.

---

6) Portfolio sizing and allocation catalog (Sizers)

This section documents the Sizers currently present in your `allocators.py` library.

6.1 RiskParityVolTargetSizer

Concept: Cross-sectional inverse-vol weighting plus a portfolio volatility target, drawdown scaling, and an approximate gross leverage cap. The code explicitly notes this is not covariance-based “true risk parity,” but an inverse-vol approximation.

Default parameters

- `lookback` : 60
- `ema_alpha` : 0.2
- `target_portfolio_vol` : 0.15
- `max_weight` : 0.30
- `leverage_cap` : 1.5
- `vol_floor` : 1e-4
- `annualization` : 252
- `allow_short` : True
- `dd_soft` : 0.10
- `dd_hard` : 0.20
- `min_notional` : 0.0
- `debug` : False

When to use

- Multi-asset portfolios where you want to reduce exposure to the most volatile assets and constrain concentration.

6.2 DynamicKellySizer

Concept: Sizes exposure using a “safe Kelly” fraction with guardrails, drawdown scaling, and optional gross leverage clamping. It can pull `kelly_p` and `kelly_b` from the strategy if provided; otherwise it uses defaults.

Default parameters

- `default_win_rate` : 0.55
- `default_win_loss_ratio` : 2.0
- `kelly_fraction` : 0.5
- `max_leverage` : 2.0 (interpreted as max fraction of equity per asset)
- `gross_leverage_cap` : null
- `allow_short` : True
- `dd_soft` : 0.10
- `dd_hard` : 0.20
- `min_notional` : 0.0
- `debug` : False

When to use

- When you have a credible estimate of win rate and payoff ratio and want sizing to respond to edge (with safety brakes).

6.3 EqualWeightSizer

Concept: Equal-weight per asset, with optional shorts, optional pyramiding, minimum notional filter, and drawdown scaling. Long/short mechanics are explicitly handled to avoid accidental short pyramiding.

Default parameters

- `weight` : 0.1
- `allow_pyramiding` : False
- `allow_short` : False
- `min_notional` : 0.0
- `dd_soft` : 0.10
- `dd_hard` : 0.20
- `debug` : False

When to use

- Baseline portfolio research; easiest to interpret.

6.4 MaxPositionsEqualWeightSizer

Concept: Equal-weight sizing with a **global maximum number of open positions**, plus drawdown scaling. Also includes “pending-order aware” logic to reduce the probability of exceeding max positions in the same bar.

Default parameters

- `weight` : 0.1
- `max_positions` : 5
- `allow_pyramiding` : False
- `allow_short` : False
- `min_notional` : 0.0

- `dd_soft` : 0.10
- `dd_hard` : 0.20
- `debug` : False

When to use

- When you want diversification control and interpretable exposure constraints.

6.5 VolatilityTargetSizer

Concept: Per-asset volatility targeting with drawdown scaling and optional gross leverage clamp. Designed with correct long/short mechanics (buy covers short; sell closes long; sell from flat opens short if enabled).

Default parameters

- `lookback` : 20
- `target_risk` : 0.02
- `annualization` : 252
- `fallback_vol` : 0.5
- `allow_pyramiding` : False
- `allow_short` : True
- `gross_leverage_cap` : null
- `min_notional` : 0.0
- `dd_soft` : 0.10
- `dd_hard` : 0.20
- `eps` : 1e-8
- `debug` : False

When to use

- When you want each position to consume approximately the same risk budget, particularly important in crypto.

6.6 ATRStopRiskSizer

Concept: ATR-based risk-per-trade sizing designed for strategies that conceptually place stops at ~N*ATR. Includes drawdown scaling and optional gross leverage cap; implements correct

long/short Backtrader semantics.

Default parameters

- `atr_period`: 14
- `stop_atr_mult`: 3.0
- `risk_per_trade`: 0.01
- `allow_pyramiding`: False
- `allow_short`: True
- `gross_leverage_cap`: null
- `min_notional`: 0.0
- `dd_soft`: 0.10
- `dd_hard`: 0.20
- `eps`: 1e-8
- `debug`: False

When to use

- When your entry logic is strategy-driven but you want consistent “risk at stop” sizing across assets.

---

7) Parameter editing (JSON) guidelines

A) Strategy params JSON

- Use a single JSON object: `{ "param": value, "param2": value2 }`
- Keep parameter names exactly as defined in the strategy.
- Use numeric types consistently:
  - Periods are typically integers (e.g., `30`)
  - Thresholds and fractions are floats (e.g., `0.02`, `0.10`)
- Booleans must be true/false in JSON.

Example: EnhancedDualMomentumStrategy (starting point)

```
{
  "momentum_lookback": 30,
  "short_momentum_lookback": 7,
  "rebalance_period": 7,
  "use_market_filter": true,
```

```
  "market_sma_period": 30,
  "drawdown_protection": false
}
```

Defaults and available keys are defined in the strategy class.

B) Sizer params JSON

Example: VolatilityTargetSizer (conservative starting point)

```
{
  "lookback": 20,
  "target_risk": 0.02,
  "allow_short": true,
  "gross_leverage_cap": 1.5,
  "dd_soft": 0.10,
  "dd_hard": 0.20,
  "min_notional": 0.0
}
```

Sizer defaults and behavior are defined in the Sizer class.

---

8) Recommended configuration “recipes”

Recipe 1: Cross-asset momentum selection with risk-controlled sizing

- **Strategy**: EnhancedDualMomentumStrategy
- **Sizer**: VolatilityTargetSizer
- **Use when**: you want the strategy to select winners/losers while the sizer normalizes risk.

Suggested tweaks:

- Increase `momentum_lookback` for slower regimes (e.g., 60–120).
- Keep `target_risk` modest and set `gross_leverage_cap` if you trade many assets.

Recipe 2: Breakout research with robust portfolio controls

- **Strategy**: BBSqueezeBreakout

- **Sizer**: MaxPositionsEqualWeightSizer
- **Use when**: you want to cap concentration and avoid portfolios that “pile into” correlated breakouts.

Recipe 3: Trend-following with explicit stop/take-profit structure

- **Strategy**: DualEMACrossStrategy
- **Sizer**: RiskParityVolTargetSizer
- **Use when**: you want a classical signal generator plus disciplined cross-sectional sizing.

Recipe 4: Strategy-managed risk sizing (Sizer secondary)

- **Strategy**: VortexTrendCapturePortfolio
- **Sizer**: EqualWeightSizer (or leave as a neutral/default)
- **Use when**: you accept that the strategy internally controls risk-per-trade sizing and your main goal is signal validation plus portfolio max-position constraint.

---

9) Troubleshooting and validation checklist

“No trades are happening”

Common causes:

- Not enough warm-up bars for indicators (increase date range).
- Thresholds too strict (ADX threshold too high, squeeze threshold too low, market filter always blocking).
- Sizer returning 0 because of `min_notional`, leverage cap, drawdown clamp, or vol estimation issues.

“Positions are tiny or always zero”

- For volatility/ATR-based sizing, check:
  - `fallback_vol` too high (shrinks size)
  - `target_risk` too low
  - `gross_leverage_cap` too low
  - `min_notional` too high

“Exposure is much larger than expected”

- Ensure your strategy is not internally sizing (passing `size=`) while you assume the sizer is controlling.

“Backtest looks too good”

- Validate with non-zero commission/slippage.
- Re-run on different date windows (walk-forward thinking).
- Reduce parameter degrees of freedom before iterating.

---

10) Practical extensions (how to evolve Portfolio Lab safely)

If you continue building this into a research-grade tool, the most valuable next documentation additions are:

- A definitive list of which **data provider** is used per ticker format (and any mapping rules).
- Exact definitions for **Statistics tab metrics** (formulae, annualization conventions, benchmark-relative computations).
- Trade log schema (what “Data” tab outputs: bar returns, portfolio returns, orders, positions, transactions).
- A “strategy contract” section: what a new strategy must implement to work well with portfolio sizers (e.g., avoid explicit `size=` unless intentionally bypassing sizing).

If you want, I can also produce a ready-to-paste README.md version of this manual with a cleaner “documentation site” structure (Overview, UI Reference, Strategy Reference, Sizer Reference, Examples, Troubleshooting) using the same underlying content.